NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

TRUST AND ITS RAMIFICATIONS FOR THE DOD PUBLIC KEY INFRASTRUCTURE (PKI)

by

Leonard T. Gaines

September 2000

Thesis Co-Advisors: James Bret Michael
Rex Buddenberg

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2000	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE Trust and its Ramifications for the DoD Public Key Infrastructure (PKI)	**5. FUNDING NUMBERS**
6. AUTHOR(S) Gaines, Leonard T.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	**12b. DISTRIBUTION CODE**

13. ABSTRACT *(maximum 200 words)*

In order to incorporate trust into e-commerce, public key cryptography, and basic communication, one must understand and effectively manage trust. Various Internet security protocols have attempted to address this lack of trust. However, these protocols do not incorporate the user's trust into these protocols. Computational models of trust have been developed in an attempt to automate the logic, variables, and thought processes that a human performs when making a trust-decision. Due to the fact that trust is based on a subjective belief, the models require the assignment of metrics to belief variables or attributes that will have value when evaluating trust. These models address the notion of trust in many different ways and both their definitions and metrics vary significantly. This thesis evaluates the various trust models. It is necessary to understand how trust is defined in each model in order to evaluate how well the operation of a system based on the model satisfies the requirements of the users. Trust models are evaluated based on their characteristics, environmental references, metrics, variables used, and outputs. This thesis concludes with the assessment of a practical application of a trust model to the DoD's PKI system.

14. SUBJECT TERMS Trust models, PKI, Computer Security	**15. NUMBER OF PAGES** 164
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFI- CATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited

TRUST AND ITS RAMIFICATIONS FOR THE DOD PUBLIC KEY INFRASTRUCTURE (PKI)

Leonard T. Gaines
Lieutenant Commander, United States Navy
B.S., University of Nevada, 1986

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

and

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2000

Author: ______________________
Leonard T. Gaines

Approved by: James Bret Michael
James Bret Michael, Thesis Co-Advisor

____
Rex Buddenberg, Thesis Co-Advisor

____
Chairman, Information Technology Group
Dan Boger

ABSTRACT

In order to incorporate trust into e-commerce, public key cryptography, and basic communication, one must understand and effectively manage trust. Various Internet security protocols have attempted to address this lack of trust. However, these protocols do not incorporate the user's trust into these protocols. Computational models of trust have been developed in an attempt to automate the logic, variables, and thought processes that a human performs when making a trust-decision. Due to the fact that trust is based on a subjective belief, the models require the assignment of metrics to belief variables or attributes that will have value when evaluating trust. These models address the notion of trust in many different ways and both their definitions and metrics vary significantly. This thesis evaluates the various trust models. It is necessary to understand how trust is defined in each model in order to evaluate how well the operation of a system based on the model satisfies the requirements of the users. Trust models are evaluated based on their characteristics, environmental references, metrics, variables used, and outputs. This thesis concludes with the assessment of a practical application of a trust model to the DoD's PKI system.

TABLE OF CONTENTS

LIST OF FIGURES

ACRONYMS

ANSI	American National Standards Institute
API	Application Program Interface
ASD(C3I)	Assistant Secretary of Defense, Command, Control, Communications and Intelligence
CA	Certification Authority
CGI	Common Gateway Interface
CMS	Communication Security Material System
CRL	Certificate Revocation List
DES	Data Encryption System
DISA	Defense Information Systems Agency
DNS	Domain Name Server
DoD	United States Department of Defense
DSL	Digital Subscriber Line
FIPS	Federal Information Processing Standard
GAO	Government Accounting Office
GUI	Graphical User Interface
IA	Information Assurance
IETF	Internet Engineering Task Force
IMAP	Internet Message Access Protocol
IP	Internet Protocol
ITSEC	Information Technology Security Evaluation Criteria
KA	Key Authenticity
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MAC	Message Authentication Code
MISSI	Multi-Level Information Systems Security Initiative
NATO	North Atlantic Treaty Organization
NIC	Network Interface Card
NSA	National Security Agency
OCSP	On-Line Certificate Status Protocol
OLAP	On-Line Analytical Processing
ORA	Organizational Registration Authority
PAA	Policy Approval Authority
PCA	Policy Creation Authority
PGP	Pretty Good Protection
PICS	Platform for Internet Content Selection
PIN	Personal Identification Number
PKCS	Public Key Certificate Standard
PKI	Public Key Infrastructure
PKIX	Public Key Infrastructure (X.509)
PMA	Policy Management Authority
POP	Post Office Protocol

RA	Registration Authority
RRQ	Recommendation Request Message
RSA	Rivest, Shamir, and Adleman
RT	Recommender Trustworthiness
SDSI	Simple Distributed Security Infrastructure
SHA	Secure Hash Algorithm
S/MIME	Secure Multipurpose Internet Mail Extension
SPKI	Simple Public Key Infrastructure
SSL	Secure Sockets Layer Protocol
TA	Trusted Authority
TAO	Tactical Action Officer
TCB	Trusted Computing Base
TCSEC	Trusted Computer Security Evaluation Criteria
TCP	Transmission Control Protocol
WWW	World Wide Web

ACKNOWLEDGEMENT

The author would like to thank Bret Michael for his invaluable assistance, guidance, encouragement, and editorial skill. His dedication to this thesis was above and beyond the call of duty. The author would like to recognize Professor Rex Buddenburg for his thoughtful advice and input. The author would also like to thank Audun Josang for his comments. Without the help of these gentlemen, the thesis would not have been possible.

I. INTRODUCTION

A. COMPUTING PARADIGM

Trust in a system is influenced largely by people's experience using that system. As computing has progressed from mainframe computers to networks of personal computer, information systems have become more vulnerable to attack. User trust in the Internet has eroded as reports of vulnerabilities (e.g., bugs) and threats (e.g., hacker exploits) appear in the news. As the Internet continues to expand, uncertainty and risk of compromising confidential information from hostile or careless users also expands.

The Web has rapidly evolved from a research tool (ARPANET) to a broadly used forum for conducting electronic commerce. Private industry and government have become increasingly dependent on the communication medium provided by the Web. The computing structure has changed from mainframes to client-server applications and peer-to-peer architectures that are networked on a worldwide basis. The migration to networked CPUs has expanded our access to communication channels, but it has also exposed us to new threats and vulnerabilities.

Computer security at one time was concerned only with the physical security of the computer, its hard drives, and other associated medium (e.g., floppy disks and tapes). The most important resource to protect was computer time. Good security consisted of ensuring that the data processing center was adequately locked and guarded. As personal computers became more prevalent, the security focus shifted to protecting files and hard drives. We have witnessed a more recent shift to protecting intelligent agents (i.e., mobile code) and mobile computing systems.

The government in October, 1967 commissioned a task force to address computer security safeguards to protect classified information from remote access, resource-sharing computing systems. That task force set the foundation for defining what was to become the trusted computing base (TCB). The requirements for a TCB were specified in a set of standards called the "Trusted Computer System Evaluation Criteria" dated December 1985. It was also referred to as the "Orange Book." Although these criteria were not

widely adopted by industry, it was the first, large-scale effort to define computing security criteria and was the first effort in establishing trust in a system.

The concepts of "trusted path" and "trusted subject" have been in use for many years. However, these systems operate in a homogeneous computing base. In today's computing environment, one can no longer depend upon a single trusted monitor. Our growing reliance on the Internet to exchange information requires us to deal with heterogeneous computing bases with multiple computers and security policies. As a result, computer security had to again shift to meet new challenges that resulted from networked systems.

Networking systems created a number of new threats and vulnerabilities. Computer systems are now vulnerable from attack from multiple areas. Attacks can be performed internally from the Intranet or externally through the Internet. Physical security can no longer protect computer assets. The same channels that allow workers to utilize remote access also allow attackers an opportunity to exploit those channels.

Another vulnerability is the fact that adding more computing and information systems to the existing architecture creates additional paths for information to flow. This information can now flow in multiple directions. The emergent properties of systems can result in something that was not predicted, or could result in nothing at all. Causal dependency in the new system may be difficult to predict. Additionally, messages sent within the network are exposed to many other systems and software that we have little or no control over.

The expansion of remote login capability has also introduced additional vulnerabilities. A modem typically offered one service, the ability to login. The ability to send mail and perform other tasks was channeled through this single choke point. With the advent of networking, other services were offered such as FTP, login, disk access, remote execution, and system status. These services are more complex and add to the number of things that must be addressed when protecting computer assets. (Chestwick, B. and Bellovin, S., 1994)

Networked systems also integrate numerous computer components and communication systems. Most organizations purchase items off the shelf, so they have no influence on the design and they do not possess any detailed information on those components. Users cannot know with absolute certainty what software has entered their network, or what action those components may take. Additionally, it is difficult to predict or know what can and cannot happen within any complex system and what can be done to control the behavior of that system (National Research Council, 1999).

There are a number of vulnerabilities inherent in using networked computers. To establish trust in such a computing environment, the public must be convinced that the threats associated with vulnerabilities can be controlled or prevented. Unfortunately all of the vulnerabilities have not been identified, nor can they all be controlled.

B. VULNERABILITIES

Virus attacks have increased the risks associated with being connected to the Internet and as a result they have contributed to distrust of the Internet. A virus is a self-replicating computer program. A virus is often malicious code embedded in an executable program that can delete files, interfere with memories and slow down processing speeds. It is usually spread through infected floppy disks, attachments to e-mail and can be attached to downloaded programs. The first two known major malicious virus attacks occurred in late 1987. (Kabay, M., 1996) This attack was spread through a university computer lab via infected floppy disks. Today, however, many viruses are designed to take advantage of networks and connections to the Internet.

Today there are over 46,000 different viruses. The costs of a virus attack vary, although an USA Research study reported costs at $800 per PC infected. (Kabay, M., 1996) Some attacks have resulted in costs of over one million dollars. Although virus scanners are more prevalent than ever before, the people creating these viruses are becoming more sophisticated.

Some viruses now have stealth capabilities. Generating a checksum based on the bytes in a file is one way antivirus products detect viruses. Changing anything in the file will change the checksum. Some stealth viruses are able to compress the original file and

insert themselves, so the file is the same size as the original file. Other stealth viruses are able to circumvent the checksums by attacking low-level I/O routines that are used to compute the checksums. The viruses are able to remove evidence of their presence before the checksum algorithms are computed; thus the checksum matches with the original value.

Another way antivirus programs operate is to detect the viral code themselves via pattern matching. The programs search for virus signatures or a unique sequence of code. Virus creators have created polymorphic stealth viruses to make detection more difficult. These viruses encrypt their signatures. When they are executed, an unencrypted portion of the code decrypts the rest of the code and loads itself into memory. Other polymorphic stealth viruses have code that can be run regardless of the sequence in which it is executed. These viruses can randomly alter the sequence of code. They then compress the program thereby changing their signature. The receiving computer must load the compressed file into memory to execute them; it is usually too late at that point to detect the virus.

Antivirus software is constantly being updated to address the increasing sophistication of virus attacks, but unfortunately, they can only react to attacks. When new viruses spread on the Internet, they can propagate rapidly before antivirus software can detect or eliminate them. As a result, people must be very careful about opening up e-mail attachments, or downloading any programs from the Internet.

Worm attacks have also adversely affected people's perception of security on the Internet. A worm is also a self-replicating program, but unlike a virus it does not need a host to propagate, it is designed to spread on its own. The first worm attack occurred in December 1987. It was the Christmas Tree Worm, which exploited the mail networks on the ARPANET.

The most famous worm attack was released by Robert Morris, a graduate student at Cornell University. It attacked VAX computers running 4 BSD UNIX and SUN Microsystems Sun 3 workstations. It exploited vulnerabilities in the mailing protocols of these systems, specifically sendmail and finger programs. The worm infected the host

computer, read its mail directory and then sent itself to everyone in that directory. Although this was not intended as a malicious program, it consumed an enormous amount of processing time on the computers it did infect. Additionally, the time and manpower necessary to delete the worm and install patches was considerable.

Worms can be very dangerous because they use networks and the Internet to spread themselves. They do not have to infect any programs, they just read a computer's e-mail directory and mail themselves to everyone in the directory. A recent worm attack against the Marine Corps Headquarters interrupted computing processes for an entire day. Antiviral software can defeat most worm attacks, but like viruses, they are reactionary.

People must be very careful when downloading programs from untrusted sources. The programs may contain a Trojan horse. A Trojan horse is an innocent looking program that has additional malicious functions. An example is a space fighter program that when installed also copies files and e-mails them to another site. Trojan horses can also be used to modify files, capture passwords, and install backdoors. Trojan horses are installed in popular games or programs that are passed around offices via floppy disk or e-mail. Common Trojan horse programs are detectable by antiviral software, but some like the new Back Orifice program contain polymorphic stealth technology and are difficult to detect.

A back door is an undocumented access code or procedure for accessing information. A programmer can intentionally install a back door, or it can be an error in the program that allows a user special privileges. One form of a back door inserts a user name in the SAM (password) file with either no password, or a prearranged password. The hacker accesses the computer using the new user name and usually gains root access to that computer. Back doors are usually inserted via Trojan horse programs, but they have also been found in commercially provided software packages as well. Robert Morris utilized a back door in the debug option for the sendmail program in the UNIX system to launch his worm attack. They can be installed in software during manufacturing or distribution. Trap doors are very difficult to detect. (Kabay, M., 1996)

The very architecture that makes it possible for multiple computers to communicate with each other also makes them vulnerable to packet sniffing. On Ethernet or Ring local area networks (LANs), when one computer transmits, all of the computers on that LAN will receive the message. The receiving computers read the header information on the packets to determine the destination address. If the destination address is theirs, they accept the packet; otherwise they discard the packets. A sniffer is a devise that listens to all of the traffic along the LAN and instead of discarding packets; it captures and copies them. By putting a network interface card (NIC) card in promiscuous mode, all traffic along the LAN can be read. This vulnerability is used by hackers to gain information on passwords, credit cards, and other private information.

There are a number of other vulnerabilities that exist as a result of networking computers such as the following: denial of service attacks (flooding), impersonating individuals and web sites, and wiretapping transmissions. It is important to note that although there are a number of vulnerabilities present as a result of networking, not all pose a threat. Some vulnerabilities cannot be exploited by hackers and others do not pose a significant threat even if they are exploited (the probability of attack and the cost to fix the vulnerability far exceeds the limited amount of damage a hacker could do if the vulnerability was exploited).

C. HACKER THREATS

Hackers are people who abuse information systems or use them to commit criminal acts. Hackers are able to break into systems utilizing the resources of the Internet to take advantage of flaws in programs and computer operating systems. It is difficult to ascertain the extent of damage done by hackers, because many companies will not report breaches in their security. They do not want to undermine public confidence in their computing systems (financial institutions are required by law to report such attacks and any related losses).

Hackers have gained a great deal of notoriety in the press and in movies. Often hackers are depicted in a positive light as gifted, but misunderstood individuals. Many hackers themselves feel that they are performing a service to society by exposing security

flaws (i.e., vulnerabilities) that the vendors or users otherwise would not correct. What hackers do not mention are the millions of dollars of damage that they do once these flaws are discovered and that about 1.8 billion dollars were spent this year by both public and private organizations trying to protect their assets (Thomas, S., 1999). Hackers directly contribute to the public's distrust of the computer as a secure mechanism for facilitating information flow. Hackers have exposed the public to the realization that there are security vulnerabilities in the Internet and computing systems.

A common misconception is that hackers are ingenious programmers. In reality these types of people are the exception. Most hackers use programs and vulnerabilities that were designed or exploited by other people. Instruction on how to break into Window's NT systems, web servers and modems are available on the Internet at hacker web sites. These even include downloadable code that can be used in cookbook fashion. At one site www.inforwar.co.ak/articles/simpnt4.htm, Lopht crack (a program for cracking passwords), getadmin and crack4.exe (used to insert a user account into the SAM file) are readily available for download. Back Orifice 2000 can be downloaded from the Cult of the Dead Cow homepage.

Back Orifice 2000 is a remote network assessment tool that was developed by the Cult of the Dead Cow for use on Microsoft NT operating systems. It consists of two parts: a client and a server. The server is installed on a target machine via a Trojan horse. The client residing on another computer then uses the Back Orifice program to take control of the server. Back Orifice can perform the same functions as most remote administrative tools such as upload and download files, delete files, change permissions and run programs, but it can also capture keystrokes and passwords, turn microphones and cameras on and off and perform a number of other malicious tasks. Back Orifice is difficult to detect because it contains a random number generator that gives it a different signature every time it is compressed. Back Orifice can be installed on a server computer without the owner ever knowing.

Hackers rely upon four conditions to accomplish their purposes: Internal security defects, misuses of legitimate tools, improper maintenance or ineffective system design

or detection capabilities (Hale, R., 1998). An example of improper maintenance, or ineffective system design is Window's NT. Window's NT is shipped with a number of defaults that if left unchanged present major security threats. An administrator must make numerous changes to the default setting to make it secure. Additionally, there are numerous patches and service packs that must be installed to prevent hackers from utilizing known vulnerabilities. System administrators must be extremely knowledgeable and have the staff and resources to effectively implement Window's NT and combat hackers. Unfortunately this is rarely the case.

One of the most exploited vulnerabilities is the buffer overflow attack. A buffer overflow occurs when the data input from a program is longer than the buffer can handle. The excess data overflows the stack. When the buffer overflows, the attacker can overwrite the internal stack space inserting commands. These commands can gain root-level access to the system.

Another common attack is to exploit weaknesses in protocols. IMAP (Internet Message Access Protocol) attacks take advantage of the fact that IMAP and POP3 need to run as root on UNIX systems. Attackers force the system to run arbitrary commands in order to gain root access and gain superuser privilege.

Hackers also can exploit CGI (Common Gateway Interface) scripts. If web servers are not configured to check all values that are input into a program for special characters and length, it is vulnerable to attack. Hackers can input special characters called meta- characters ('$;<*>&, etc.) that can command the web server to mail the password file back to the attacker.

Most hackers will break into a number of computers before starting their attacks. They will break into one computer and will then use that computer to break into another computer. The more computers they use, the harder it is for authorities to trace. Hackers are now taking advantage of high speed digital subscriber line (DSL) connections used by home computers. These connections are almost always on, so their IP addresses are virtually fixed. This allows hackers to easily find and exploit these connections. With the introduction of DSL and cable modems, hacking into home computers is becoming

more common. What many people do not realize is that it as easy to connect to their computer as it is for them to connect to the Internet. As hackers continue to exploit high-speed connections in private residences, identity fraud will become more common.

The more sophisticated hackers have become adept at address spoofing. They alter the TCP/IP packets to make it appear that the message came from within the company's network. The attacker hopes that the fake IP address will defeat simple security systems that employ source address filtering. To prevent this attack, a firewall must be configured to discard packets with an internal source address that arrives on an external interface.

Once hackers gain access to a network, they can place sniffers on the net to capture all traffic, including passwords. This allows them to capture, modify, delete and replay any message that travels over the network. Additionally, once they have compromised the network, or a machine within the network, they can masquerade as someone else. The hackers can send messages that appear to come from authorized sources. They can also intercept messages and send fraudulent acknowledgements. This can create huge problems within an organization when you are never certain with whom you really are communicating.

Hackers have eroded people's trust in the Internet and in computer security. As soon as products are released, security vulnerabilities are being reported. In some cases patches to fix security flaws actually create more vulnerabilities. Hackers are constantly probing new products and developing new methods of attack. Unfortunately, computer security is often reactionary, with response occurring only after new attack methods are discovered. Additionally, security concerns must be weighed against ease of use in most organizations. Thus the most secure implementations are usually not implemented. As a result, any machine that is connected to the Internet is potentially vulnerable to attack.

D. TRUSTWORTHY SYSTEMS

Although the Internet is an untrustworthy entity, there are measures that an individual organization can take that will ensure a certain level of trust in that entity. Computer security is often defined in terms of confidentiality, authentication, integrity,

and availability. Confidentiality ensures that information within a computer or transmitted can only be read by authorized personnel. Authentication ensures that the entity sending a message in correctly identified. Integrity means that only authorized personnel can modify computer assets or transmissions. Availability ensures that computer assets are available when needed. There is no single computer security mechanism that can provide total security, but a combination of mechanisms can be used to provide a reasonable level of trust in the system.

Confidentiality is generally performed through cryptography. There are a number of different cryptographic algorithms that can be used including Triple DES, PGP, Blowfish, RSA and Cast-128. It is important however to choose a strong algorithm that uses a large key.

There are a number of authentication techniques, but all contain elements that verify something a person knows, something a person has, or something a person is. The most common authentication method is the password, which is something the user knows. Smart cards, badges, reply calculators and tokens are something the user has. Often authentication schemes will combine knowledge a person has with an object he possesses, such as an ATM card with a PIN number. Biometrics makes use of a person's physical characteristics, such as fingerprints, retinal scanners, voice authentication, and pressure-sensitive signature pads.

Integrity is usually accomplished via a digital signature. A digital signature combines public key cryptography with a hash algorithm. A hash is a cryptographic code that is produced by running a message through a hash algorithm. The hash value is appended to the message and transmitted. The receiver of the message verifies the hash by putting the message through the hash algorithm again. If the message is modified in any way, the hash value will not match the hash value sent with the message. To ensure that the hash value is not modified in any way, it is encrypted with the sender's private key. The receiver uses the sender's public key to decode the hash and then verifies the decrypted hash against the message's hash. A digital signature also provides some level of authenticity as well as nonrepudiation, because the private key identifies the sender.

Availability depends upon the bandwidth of the network, the number of machines on the network, and its configuration. To prevent denial-of-service attacks from the Internet, a firewall can be installed that will prevent the ping of death (i.e., message flooding) and other attacks.

The degree to which a system can provide authenticity, integrity, confidentiality, and availability determines the level of risk associated with that system. The extent to which a system is secure helps establish the level of trust afforded to a system. Unfortunately, it is extremely difficult to judge a system's security posture unless one is intimately familiar with it, or it has been evaluated by a reputable outside agency. The Internet is a heterogeneous computing environment with differences in security policies among the information systems connected to it. This makes it difficult to evaluate trust among entities on the Internet.

In electronic commerce, trust has become a key factor in ensuring a business' competitive reputation. If the public does not trust a company's product or security, it will not do business with that company. In traditional (i.e., non-electronic) business transactions, there are legal and social methods to establish trust. Legal methods consist of paper contracts and signatures that legally bind entities to that contract. Social methods involve the methods an individual uses to determine if another person is trustworthy. These methods include personal relationships, face-to-face interaction (e.g., the individual looks nervous), and an assessment of the business environment (e.g., the business is run out of a garage instead of an office building). Unfortunately these methods are not available on the Internet, although digital signatures are becoming legally binding in some states.

Although an entity's security posture is a factor in people's perception of its trustworthiness, other factors also determine trust. The reliability of the information from a source has little to do with its method of secure communication. A public key certificate cannot tell you about the integrity of the person that owns the certificate. In order to incorporate trust into e-commerce, public key cryptography, and basic communication, one must understand and effectively manage trust.

E. RESEARCH GOAL

Trust is a human cognitive function. Trust modeling is an attempt to emulate the way a human assesses trust. There are a number of trust models that represent attempts to define and assign metrics to trust. These models address the notion of trust in many different ways and their definitions and metrics vary significantly.

Depending upon an entity's security policy and how that entity chooses to implement trust models, multiple levels of trust may have to be addressed. Additionally, the interpretations of trust vary among computing bases, domains, and applications. This makes it difficult to convey trust between entities on the Internet.

This thesis will assess various models concerning trust. We will first define the concept of trust. We will then compare and contrast various trust models by evaluating their characteristics, environmental references, metrics, variables used, and outputs. The paper will then apply these concepts of trust to the Department of Defense's (DoD's) Public Key Infrastructure (PKI) system.

One of the challenges in conducting this research is that the DoD's PKI is constantly changing. The system has yet to be implemented and there is no definitive policy at this time. We therefore will evaluate the latest policy, which may differ from the final approved policy.

F. ORGANIZATION OF THE THESIS

Chapter II will consist of a literature review of trust, trust modeling and trust management. This will provide a background for a subsequent discussion of the various truth models.

Chapter III will discuss the various definitions of trust. It is important to define trust before it is modeled. In many instances Internet certification protocols attempt to deal with the concept of trust, without ever defining trust. Without a formal and commonly accepted definition and identifications of the components of trust, how can a protocol effectively deal with the issues related to trust?

Chapter IV contains an analysis and discussions of the various trust models. This chapter will provide an assessment of the strengths and weaknesses of the models and their suitability for commercial application.

Chapter V contains a description of the DoD's PKI system. Chapter VI consists of a case study applying a truth model to the DoD's PKI system.

Chapter VII discusses whether subjective inputs can be incorporated into formal analyses. Chapter VIII summarizes my conclusions and recommendations. It also contains a section on future research directions.

THIS PAGE INTENTIONALLY LEFT BLANK

II. LITERARY REVIEW

A. SIGNIFICANT LITERATURE ON TRUST DEFINITIONS

Trust has been discussed in philosophical terms for centuries. The great philosophers, clergy and statesmen wrote extensively about trust and its implications on man and society. Early literature relating trust applications to modern computing focused on the concept of a trusted computing base (TCB). The TCB was concerned with reducing or eliminating threats to the compromise of classified information. With the advent of the Internet, the concept of trust in an inherently untrustworthy environment has become more important, as discussed in Chapter 1. To understand how important trust is to networked applications, one must first be able to define trust.

Ed Gerck in his paper "Towards Real-World Models of Trust: Reliance on Received Information" attempts to define trust and relate it to real-world models.

The author's first premise is that trust must first be understood before it can be modeled. The author bases his definition of trust after Shannon's definition of information. Shannon stated that information is what you do not expect. Modifying this premise, trust can be defined as "trust is that which is essential to a communication channel, but cannot be transferred from a source to a destination using that channel." (Gerck, E., 1998, p. 3)

The author proposes a central abstract definition of truth. This notion of truth is the seed concept for all other definitions of trust. From the abstract definition, explicit definitions are derived by applying a stance and an observer. The explicit definitions are applied to real-world scenarios. This leads to a number of different definitions of trust, but since they are all derived from the same abstract notion, they can be combined. This would allow for more complex notions of trust that incorporate different aspects of trust.

The author also discusses the mathematical properties of trust. Trust has the following mathematical characteristics:

- It is not transitive

- It is not distributive
- It is not associative
- It is not symmetric.

Based on the mathematical characteristics he proposed a truth model. This model was based on subjective determinations of non-zero values assigned to the characteristics. The author is in the process of developing a "trust algebra" using Grassmann's Algebra (Hermann Grassmann developed the theory of basis and dimension for finite-dimensional linear spaces or Universal Algebra (Fearnley-Sander, D., 1982)) that can represent trust.

In his appendix the author explores the concepts of process trust and social trust. Process trust deals with the technical use of the word trust. It applies to concepts such as digital certificates, digital signatures, hardware, and software. Social trust is concerned with the emotional concepts of trust. The author evaluates six definitions of trust based upon "process trust" including those from NSA, X.509, ABA Digital Signature Guidelines, ABADSG II, PGP, and a dictionary definition of trust. Ultimately, the author concludes that a social model of trust must be used in conjunction with process trust to adequately represent a communication trust model.

The author also discusses the notion of distance. In quantitative terms, how close the input data is to data that can be trusted. Metric functions can be used to define the distance between point x and y, or d(x,y). The following mathematical rules must be met:

- $d(x,y) = d(y,x)$
- $d(x,y) + d(z,y) \geq d(x,y)$
- if $x = y$ then $d(x,y) = 0$
- if $d(x,y) = 0$ then $x = y$

This concept allows us to define how close or far some input is from being trusted, and can even be used to provide paths to move "closer to the truth."

The last part of the article is a discussion on whether unique names are needed on the Internet. The author sites work by Frege, which states that a name has two independent components. The first component is reference (the symbol itself, string) and

sense (the symbol's meaning). The name's sense represents the name's truth conditions and the name's reference represents the name's truth-values. Thus an unlimited amount of entities can share the same reference, but each will be uniquely identified by their sense. Since it is not possible to derive meaning from a name, it is not necessary to have a unique name, as they are all meaningless. The link between reference and sense is provided by "proper trust". The author describes this concept in more detail in a system called TSK/P (Trust, Semantics, Keys over Pragmatics). (Gerck, E., 1998)

Adrian McCullagh, in "The Establishment of 'Trust' in the Electronic Commerce Environment" wrote another article that discusses the definition of trust. The author proposes that trust in electronic commerce relies upon the interaction of four components: technical trust, behavioral trust, product trust, and legal trust.

Technological trust involves the classification of information and the level of security mechanisms. TCSEC, ITSEC and common criteria are examples of technological trust. Behavioral trust involves the way that different societies view trust. Product trust is usually achieved through marketing. Legal trust involves the need to establish an adequate legal framework to establish trust.

The author also proposed some of the basic properties of trust. Many of these are the same properties outlined by Abdul-Rahman and Hailes. The properties are:

- Trust is not associative.
- Trust is generally not transitive.
- Trust is always between two entities.
- Trust in non-symmetric.
- Trust will involve either direct trust or recommender trust.

The author also mentioned work done by Fukuyama in "Trust – the Social Virtues in the Creation of Prosperity". Fukuyama grouped behavioral trust into two types of societies. The low-trust culture tended to be family oriented and trust only family relatives. People outside of their family are not trusted. The Chinese and French fall into this category.

The other society is the high-trust culture. Trust is not limited to specific families or organizations. America and Japan fall into this category.

It is the author's contention that because of the societal influences and the disparate components of trust, it cannot be reduced to a mathematical equation. Cultural influences cannot be underestimated. (McCullagh, A., 1998)

Audun Josang in his article "Prospectives for Modelling Trust in Information Security" defines two types of trust. He believes that a human is trusted if believed to be benevolent and distrusted if believed to be malicious. The mental process to decide between benevolent and malicious behavior is called "free will". The agents possessing this type of free will are designated as "passionate." Therefore, trust in a passionate agent is the belief that it will behave without malicious intent.

Hardware and software are not passionate, nor do they exhibit free will, but they are trusted. This type of agent is called "rational" as opposed to passionate. Trust in a rational entity is defined as the belief that it will resist attacks from malicious agents. (Josang, A., 1997)

Alfrez Abdul-Rahman and Stephen Hailes used Diego Gambetta's definition of trust in their article "A Distributed Trust Model". Gambetta felt that trust was a particular level of subjective probability with which an agent will perform a particular action in a context that affects our own action. (Abdul-Rahman, A. and Hailes, A., 1998, p. 49)

B. SIGNIFICANT LITERATURE ON TRUST MODELS

Trust models have been used to mimic human trust, dissect trust into element parts, categorize trust, and assign metrics to trust. The designers of the trust models try to develop a system to communicate a notion of trust from one entity to another. Since trust is a subjective belief, the models assign a metric to that belief that will have value when evaluating trust. Trust modeling and its application to the Internet is a relatively new topic of discussion, and, as a result, there is very little literature on this topic. However, some very informative articles have been published.

Alfrez Abdul-Rahman and Stephen Hailes discussed their trust model in their article "A Distributed Trust Model". In this article they not only define trust, but they also discuss the weaknesses of various security models in dealing with trust. The authors contend that due to the inherent security risks associated with the Internet, users must have a means of determining the trustworthiness of entities they encounter. They state that current security systems have trust built into the systems, but they fail to manage that trust effectively.

The authors address some of the problems associated with managing trust and propose a distributed trust model. Some of the problems addressed were the following:

- A Trusted Authority (TA) can never be good enough for everyone in a large organization. As the organization grows, the TA's credibility declines and the level of uncertainty with respect to recommendations increases.
- Common assumptions of transitivity inherent in most authentication protocols do not hold.
- Transitivity may be true if certain conditions (outlined in the paper) are met. This procedure is called conditional transitivity.
- In a large system, it is almost impossible to know every entity in the organization, much less have enough intimate knowledge of the entity in order to develop a trust relationship. This situation demands the need for recommendations of trust.

The trust model developed was based upon two types of relationships. In a direct trust relationship, Alice trusts Bob. In a recommender trust relationship, Alice trusts Bob to give recommendations about other entities' trustworthiness. The model specifies trust categories, trust values, and a protocol for judging recommendations from other agents. It also supplied an ad hoc algorithm for determining a level of trust. (Abdul-Rahman, A. and Hailes, A., 1998)

Andrew Myers and Barbara Liskov developed a trust model in their article "A Decentralized Model for Information Flow". This article is concerned with the downloading of distrusted code, and how the user can control how that code transfers shared information to others. At the heart of the article was the question, how does one

control information propagation, yet not overly restrict the user? A trust model was needed to perform that function.

If user A was allowed to read user B's file, B cannot control how A distributes the information it has read. Systems exist to control this propagation, but they tend to restrict necessary information flow. The propagation of information becomes more of a problem, as the enterprise grows larger and more complex. In this environment trust in unattainable.

The authors proposed a model that allows users to control the flow of information without the constraints usually imposed by a multilevel security system. The authors' model allows users to declassify data that they own. When data is from several sources they all must declassify the data. Users declassify via a central authority. One of the goals of the model is that the amount of trust that the user has to place in the system be limited. (Myers, A. and Liskov, B., 1997, pp. 129-142)

Daniel Essin focuses on trust within health care organizations in his article "Patterns of Trust and Policy." The emphasis of the paper is on the fact that trust in not static: it must be judged in context. There are different layers of trust based on differences in knowledge, reputation, or the stake that the reviewer has in the outcome. The paper focuses mainly on the socio-technical aspects of trust instead of traditional access control measures.

He devised a trust model based upon context, identity, reputation, capability, and stake. Unfortunately, he only proposed a partial model and did not assign any values to the model, nor were there any measures of the trust values generated by the model. The model did however, generate some good points and it identified a number of variables that could be assigned to trust. (Essin, D., 1998, pp. 38-47)

Michael Reiter and Stuart Stubblebine wrote an article titled "Authentication Metric Analysis and Design" in which they evaluate a number of trusted models based on specific metrics that they developed. In doing so, they pointed out weaknesses in many of the common practices associated with trust and authenticity. The authors developed a number of principles they felt are essential in designing metrics for authentication.

Three main areas are evaluated: The meaning of the values output by the metric, the extent to which the metric could be manipulated by malicious behavior and whether the metric can be practically applied.

Metrics proposed by Beth-Borcherding-Klein, Maurer, Reiter-Stubblebine, and Zimmerman (PGP) are evaluated against the proposed principles. Additionally, the authors developed their own metric that they feel meets all of their principles. (Reiter, M. and Stubblebine, S., 1999, pp. 138-158)

Audun Josang developed a trust model that utilizes an opinion model and a process he calls subjective logic that consists of a set of algebraic operators. His model can be applied to a number of applications including authentication, security evaluation, artificial intelligence, and e-commerce. He has written extensively on the subject.

In his article "An Algebra for Assessing Trust in Certification Chains," he proposes an algebra for trust that can be used to determine the authenticity of received keys.

An opinion is a function of degrees of belief, disbelief, and uncertainty. Mathematically, belief is defined by the expression $b + d + u = 1$, where b, d, and u represent belief, disbelief and uncertainty. Another expression for belief is $w = \{b,d,u\}$.

The algebra for trust is based upon subjective logic. Subjective logic can be called a calculus for uncertain probabilities. The user can use the algebra to derive a combined opinion about two separate statements. The user can calculate an opinion about a statement based upon another user's recommendation. A consensus can also be calculated from the opinion of two agents concerning the same statement. The result is a user can calculate a trust value on a statement utilizing a certificate chain.

An authentication measures trust in three areas. The recipient of the certificate must have an opinion about the key authenticity of the key used to certify (key to user binding). The recipient must have an opinion about the certifier's recommendation trustworthiness. Finally, the certifier must give the recipient his opinion about the authenticity of the certified key. All recommendations must be based on first hand evidence only. Second hand evidence is not allowed. (Josang, A., 1999)

Josang added more elements to his opinion model in his paper titled "A Logic for Uncertain Probabilities." The article starts with a discussion of the Dempster-Shafer belief model. This model was first proposed by Dempster in the 1960s and was expanded by Shafer in 1976. It starts by defining a "frame of discernment" which delimits a set of possible states of a given system, exactly one of which is assumed to be true at any one time. The elementary states in the frame are called atomic states. The powerset of the frame contains the atomic states and all possible unions of the states, including the value of the frame itself.

An observer who believes that one or several states in the powerset are true can assign "mass belief" to those states. Mass belief on an atomic state is interpreted as the belief that the state in question is true. A mass belief in a state X does not express any belief in substates of X (atomic elements combined to form the set of X).

The author adds another element to his original discussion on subjective logic. He adds the element called relative atomicity. The idea is that when the atomicity of a state is greater, its mass belief is spread out on more substates (atomic elements), so that the contribution of each substate decreases. For example: If I assign a mass belief value of 1 (certain knowledge) to state X, and state X is composed of the atomic states X1, X2 and X3, I know one of the atomic states is true, I just do not know whether it is X1, X2, or X3. If there were ten atomic states, the mass belief would be spread out over a greater number of states.

Mass belief assignments and relative atomicity are used to assign a probability expectation for a particular state. The probability expectation formula can be used to determine atomic states and superstates. (Josang, A., 1999)

In the article "Trust-Based Decision Making for Electronic Transactions," Josang applied utility theory to his trust model. He combined the probability expectations of events with positive or negative utilities that could result from that event. Each agent could attach a utility to an outcome of a transaction.

If the expected utility of a transaction is positive, the transaction can be executed, provided the risk is acceptable. If the expected utility is negative, then the transaction

should not be executed. The author also shows how utility theory can be useful in evaluating different certification chains.

By converting trust into a probability expectation value, formulas used with standard probability theory could now be applied to trust. (Josang, A., 1999)

In his article "Prospectives for Modelling Trust in Information Security", he discussed four formal models of trust and their strengths and weaknesses. The first model is BAN-Logic. BAN-Logic is primarily used to verify the correctness of security protocols. The general principle of BAN-Logic is to transform the steps of a security protocol and its assumptions into logical formulas. The formulas are then manipulated by a set of logical rules to determine the conclusions of the protocol. It can then be determined if the conclusions corresponded to the specified purpose of the protocol.

The second model is the BBK-Scheme. The model consists of a method for extracting trust values based upon positive and negative experiences and it derives new trust values from existing ones within a network of trust relationships. Each estimated trust value from the BBK-Scheme corresponds to a specific trust class or task.

The third model is the Shared Control Scheme by Simmons and Meadows. The simplest shared control schemes are unanimous consent schemes in which everyone must agree before an action can be performed. The model determines the additional trust needed for entities A, B, and C to act in concert in order to execute a task. (Simmons, G. and Meadows, C., 1995)

The last model is trust from evaluation assurance. Security evaluation is based upon a set of evaluation criteria graded by independent accredited evaluation labs. A successful evaluation leads to a certification stating that a specific assurance level was attained. This level reflects the degree to which the system can be trusted. (Josang, A., 1997)

Josang's paper "A Subjective Metric of Authentication" also discusses the weaknesses of other trust metrics and models, including the following: PGP, BBK model, the Mauer model, and the Reiter-Stubblebine model. He states that they all represent trust as a discrete or as a continuous parameter in the range [0,1]. He felt that discrete

models were insufficient in that they only provide a small set of possible trust values. He felt that the continuous and probability-oriented models did not accurately represent human cognitive phenomenon. He stated that probability could not reflect the conditions of ignorance and uncertainty. (Josang, A., 1998

C. SIGNIFICANT LITERATURE ON TRUST MANAGEMENT

Trust management has a number of definitions. Some believe it is the process of translating a trust model into a practical application by combining trust variables associated with authentication with those of integrity and confidentiality. Others believe it is a system for protecting open, decentralized systems by analyzing, codifying and managing trust decisions.

Yang-Hua Chu, Joan Feigenbaum, Brian LaMacchia, Paul Resnick, and Martin Strauss defined trust management in their paper "REFEREE: Trust Management for Web Applications." They argue that trust management is a system for deciding whether a requested action, supported by credentials, conforms to a specific policy.

Their paper describes the REFEREE trust management system. The paper begins with a discussion about the inherent dangers of trust on the World Wide Web. There are mechanisms to deal with specific portions of trust on the web, but there is no one system that combines all of the portions together. The REFEREE trust management system attempts to do just that.

REFEREE examines a requesting action, accesses its module databases and extracts the module containing the policy and interpreters to evaluate the action. The modules can request additional information of statements from Platform for Internet Content Selection (PICS) labels, or it can request assistance from additional modules. The modules decide if the action conforms to the policy. If so, the action is performed. If not, the system will not allow the transaction and will provide an explanation why it cannot. (Chu, Y., and others, 1997)

Yang-Hua Chu wrote his thesis "Trust Management for the World Wide Web" on the REFEREE trust management system. The goal of the thesis was to describe the lack of trust in the World Wide Web, and then propose a trust management infrastructure that

could address that lack of trust. Chu described four major components of a trust management system: the metadata format, the trust protocol, the trust policy languages, and the execution environment. The REFEREE system was designed to incorporate these four components.

For a given user request, REFEREE invokes the appropriate user policy and interpreter module and returns to the host application an answer of whether or not the request complies with the policy. The basic computing unit is a module. It is an executable block of code that processes the input arguments, compares the input to policies, and outputs an answer. The module consists of a policy and zero or more interpreters. Modules can delegate tasks to other modules if necessary. Modules can also be easily added or deleted. They are contained in a module database that cross-references the requested action with the appropriate module and interpreter.

REFEREE is a good trust management system in that it is one of the first to combine all of the categories of trust management into one system. The other system, Microsoft's Authenticode also combines all of the categories into one system, but its application is limited. It does not have the inherent flexibility of the REFEREE system. (Chu, Y., 1997)

Brian LaMacchia also worked on the REFEREE trust management system. In his article "The Digital Signature Trust Management Architecture," he expands on some of the internal architecture of REFEREE, focusing on the content of the internal and external application programming interfaces (APIs).

The author states that there are six design goals that must be met to achieve a successful trust management architecture. They are:

- The architecture should be general purpose. It should not be restricted to policies that rely only on digital signatures
- "Policy controls everything." The decisions made in a trust management system should be able to include more information than traditional systems, and it should apply to more complicated systems. An internal policy should also be able to decide which policies to apply both internally and externally

- Policy descriptions should be transferable
- Simple policies should be simply described
- Easy to implement
- Extensible: It must be capable of handling new data and formats

The internal API facilitates communication between the modules within REFEREE. Inputs consist of controlling policy, statement lists, and additional arguments. Outputs are the tri-values (i.e., true, false, and unknown) and statements. The operation of the module itself is controlled by the controlling policy. In Chu's thesis, this policy was an element of the module itself. Although it could be an overall policy controlling the actions of all of the modules to derive and answer.

The external API communicates between the user application and the REFEREE trust management system. Application inputs consist of action, policy database, interpreter database, statement lists, and additional arguments. Outputs are also tri-values and statements. The action, statement list, and additional arguments define the context of the question. The type of action is mapped to a policy database and interpreter.

The remainder of the article discusses the syntax and modules needed to execute the digital signature architecture. Additional modules are needed to validate the signature and certificates and a decryption system to check the hash. (LaMacchia, B., 1997)

Rohit Khare and Adam Rifkin discussed trust management for Web-based applications in their article "Trust Management on the World Wide Web." The authors believe that trust management is a new philosophy for codifying, analyzing, and managing trust decisions. They believe trust management asks the question "Is someone trusted to take action on some object?" (Khare, R. and Rifkin, A., 1998, p. 2)

Their discussion is from a global perspective of managing trust for all Web applications. The authors stated that the first step in managing trust on the Web is to develop a way to identify all of the principals (entities involved). They agreed that digital certificates utilizing PKI was one way to accomplish this, however they pointed out some of the limitations of the PKI.

There are two new PKI proposals that the authors argue are better suited to trust management. Both are still in their infancy. The first is Simple Distributed Security Infrastructure (SDSI) and the second is Simple Public Key Infrastructure (SPKI), which is currently being merged with the SDSI 2.0 draft. These systems issue application-specific certificates that specify exactly what the key is authorized for. Both systems construct a trust chain that loops back to the user, and both systems use real-time certificate validation.

The second step is to associate access limits with each element of the system. The authors propose using external metadata labels. These labels would be bound by URL to a specific Web resource. These labels would be in a PICS format.

The third and final step is to specify the authorization decisions according to some policy. The authors agree that the REFEREE trust management system will achieve this goal. Additionally it will determine a trust decision based upon a target, a principal, a proposed action, and the policy stated. (Khare, R. and Rifkin, A., 1998)

Rohit Khare and Adam Rifkin also discussed the attributes a trust management system must have in their article "Weaving a Web of Trust." The article touches on principles, principals, policies, and pragmatics that must exist in a trust management system.

The authors describe the following principles:

- Be specific – who is the trusted entity(s), and what exact actions do we trust them to perform. Without quantifying the bounds of the relationship, you cannot expect the bounds to be of any value. It is difficult to be specific given the tools available today.
- The second principle is trust no one but yourself. All trust begins and ends with oneself. Any truth decision should be logically derived from the axioms that one believes. The PKI system requires trust in the certification authority, whereas SPKI and SDSI loop the trust chain back to the user.
- Be careful – rigorously justify all trust decisions in your application.

There are three principals: people, computers and organizations. People are identified by their names and can make trust assertions using digital signatures and

certificates. Computers can mechanically verify their data transmissions, thereby vouching for the computer's IP address. Organizations can represent groups of people or computers with certificates.

Trust management systems such as REFEREE take as input a subject, action, statements about the subject and match that to a module containing the corresponding policy. For each action, there are specific policies that govern which statements are valid.

The authors discuss three types of approaches to framing policies. The first is principal centric policies, which tend toward a policy that only certain people can be trusted. The policy checks the clearance of each principal to determine if that principal can perform an action on an object. The other approach is object-centric policy. Handles, tokens, combinations, and cryptographic keys are the essence of object-centric policy. A principal must have a trusted object that represents permission to execute actions on another object. The final policy type is an action-centric policy. This policy states that only certain actions are trusted and it checks to ensure that any action taken by a principal on an object is approved. (Khare, R. and Rifkin, A., 1997)

In his article "The Troubling Truth about Trust on the Internet," Richard Hombeck pointed out the weaknesses of the Microsoft Authenticode trust management system. He felt that Authenticode had some interoperability problems.

Microsoft developed Authenticode for use in Internet Explorer only. If Microsoft's browser is not used, then Authenticode will not work. Another problem with Authenticode is the fact that the user has to trust that the Certificate Authority (CA) has properly identified the developer. This task is made even more difficult by the ease with which people can now forge driver's licenses, passports, and birth certificates. Authenticode places all of its security on the validity of public key cryptography, instead of analyzing the downloaded code. Authenticode is not a barrier to malicious software. Additionally, Authenticode has no certificate revocation list (CRL) verification. (Hombeck, R., 1998)

D. SIGNIFICANT LITERATURE ON PUBLIC KEY INFRASTRUCTURE

The subject of Public Key Infrastructure (PKI) is relatively recent. The subject is covered in most computer security books, but the discussion is usually very superficial. There are, however, a number of articles and books that discuss the various aspects of PKI in detail.

One of the best sources of information on PKI and digital signatures is *Secure Electronic Commerce: Building the Infrastructure for Digital Signatures and Encryption* by Warwick Ford and Michael Baum. The book provides a great overview of the entire PKI system, including the legal aspects of cryptography. (Ford, W. and Baum, M., 1997)

Anthony Hansen also provided an excellent description of the PKI system in his thesis "Public Key Infrastructure (PKI) Interoperability: A Security Services Approach to Support Transfer of Trust." In his articles, he discussed functional and technical interoperability in PKI. (Hansen A., 1999)

A paper titled, "Public Key Infrastructure (PKI)" by Cabletron provided a good general overview of the PKI process. It also had a good section discussing the requirements of PKI. Specifically, the requirements that software programs implementing PKI must meet. The paper discussed the fact that there are no known commercial products that meet all of the necessary requirements.

It also discussed briefly the fact that the Lightweight Directory Access Protocol (LDAP) has become the defacto standard for accessing directory systems. Directory services using LDAP are ideal for storing certificates and allowing applications to retrieve those certificates.

Key backup and recovery as well as key escrow were also mentioned in the article. The paper discussed who should be responsible for performing key backup in an organization as well as whether key escrow should be used. The paper discussed trust between CAs when engaged in cross-certification as well. (Cabletron Systems, June 1999)

An article that not only described the PKI system, but its implementation as well was "Design Issues in a Public Key Infrastructure (PKI)" by Douglas Barton, Anthony

Moran, and Luke O'Connor. Their paper covers issues of key management systems. In particular the article covers implementation of a PKI system in New Zealand.

This paper mentioned a number of problems associated with the implementation, but it is difficult to judge how many of these problems are still relevant given that this paper was published in 1996. However, it is a good article in that it details the many areas that must be considered when implementing a PKI system. Areas of consideration are inter-domain hierarchy structures, revocation techniques, language choice, design methodology, standards, cryptography, and certificate chains and trust.

This paper also contains a discussion of the revocation process: the various types of revocation procedures and the strengths and weaknesses of the various revocation techniques. CRL deltas are discussed in depth. (Barton, D., Moran, A., and O'Connor, L., 1996)

Robert Booker focused on the cost benefits of an enterprise-wide installation of a PKI system in his article "Practical PKI." As PKI systems can be very complex, the author suggests that an organization is often better off installing PKI gradually in conjunction with the organizations needs.

The author suggests that organizations need to evaluate their application needs and information sensitivity and implement a practical PKI system to meet those needs, instead of investing in technical elegance. It may be necessary to deploy a limited infrastructure with limited scope and then expand as new requirements and capabilities emerge.

Additionally, the level of trust necessary within the organization needs to be defined. The organization must determine which community is supported. It must also determine if it is a closed or open system, and internal or external community. Early deployment of PKI is easier with closed systems and internal communities, but e-commerce is conducted with external communities. Another major issue in deploying a PKI system is determining the level of authentication validation necessary to accomplish the organizations needs. (Booker, R., 1999)

Warwick Ford also wrote an article, "Public-Key Infrastructure Interoperation: Some Pragmatics" which discussed trust within the PKI system. It discussed the fact that it is inherent for organizations to want to operate their own public-key infrastructures. Given this fact, is it a practical assumption that we can create a global infrastructure by simply connecting all of the local infrastructures?

Ford contends that the best chain of trust is one in which a human user is empowered to make trustworthiness decisions and to be able to inspect the certificate chain at the time it is validated. This is similar to the PGP method of trust where each user can assign a level of trust to a certificate. However, as the Internet continues to expand in size, the method used by PGP is not practical. In PGP, the user or her trusted recommenders must have knowledge of the trustworthiness of the certificate that is being verified. Obtaining that knowledge will become more difficult as the number of certificates and agencies acting as certification authorities increases. Ultimately, much of the trust evaluation and decision-making will have to be automated. An automated server will decide if it trusts a presented certificate chain by inspecting that chain, compiling trust metrics, and comparing the result against a decision table.

Ford proposes several methods to increase the public's confidence in certificates and mitigate trust dilution. Among these are: CRLs, screening or accreditation of certifying organizations, mandatory security controls including physical and personnel security, mandatory auditing, X.509 constraints, financial bonding, contractual obligations with recourse to compensation for damages, insurance and consolidation into fewer larger CAs. (Ford, W., 1997)

THIS PAGE INTENTIONALLY LEFT BLANK

III. THE DEFINITION OF TRUST

A. KHARE AND RIFKIN'S TRUST DEFINITION

Trust can have many meanings. In the previous chapter we discussed some of the significant differences in the interpretation of trust. Trust is a subjective notion. Trust entails a measure of uncertainty and can interact with, or is influenced by, a number of variables. Without a full understanding of the definition of trust as it is applied to a specific application, it is not possible to apply it properly in a trust model.

Khare and Rifkin define trust as a reflection of our belief that someone will act in a certain way based on our past history and the expectations of others. Their definition is based on an assumption that trust is applied to humans, but not machines. Their rationale is that humans are ultimately responsible (legally) for the computer's behavior.

Trust in an individual computer can be established by a number of methods. The protocols used can be tested for compliance, the hardware components can be checked, and it can be measured against a trusted computing base (TCB). Trust can also be established by a set of evaluation criteria such as the Trusted Computer Security Evaluation Criteria (TCSEC), Information Technology Security Evaluation Criteria (ITSEC), and the Common Criteria. However, when dealing with a heterogeneous computing environment such as the Internet, establishing trust is more difficult. The Internet has no trusted computing base. It is also not possible to test all of the hardware and software that a message packet might interact with while communicating from point A to B. As a result, this definition is based on the belief that trust should only be placed in people, as they are the ultimate decision makers. (Khare, R. and Rifkin, A., 1997)

B. JOSANG'S TRUST DEFINITION

Trust and security are often interrelated. The perception of a level of security is an indication of the risk that is associated with it, and thus is a measure of how much that system is trusted. Systems with low security are not trusted as much as systems with high security. Conversely, if a system is not trusted, it usually is perceived as having

poor security. This analogy can be used with computer systems, but unfortunately, this concept does not apply as well to humans. Trust in humans is typically not determined in terms of how secure they are. Although humans can be assigned a security clearance (i.e., the Admiral is cleared for top secret), the trust placed in that individual was based on the characteristics that allowed him or her to attain that level of security clearance, not that individual's resistance to coercion to divulge sensitive information.

Audun Josang defines two types of trust for the purpose of information security. He felt that it was perfectly acceptable to trust or distrust both computer systems and humans, but only computer systems could be described in terms of security. As a result, he defined trust in a computer system as "the belief that it will resist malicious attack." (Josang, A., 1999) He used the term computer systems to refer not only to physical computer equipment, but also abstract entities such as cryptographic algorithms, keys, and software.

He posits that a human is trusted if he or she is believed to be benevolent, or distrusted if believed to be malicious. There are a number of variables that can determine if an individual will behave in a malicious or benevolent manner, but in practice, gathering information on all of those variables is impossible. Thus he concludes that human behavior is always unpredictable.

He defines trust in a human as the belief that the person will cooperate and not defect (act with malfeasance). In his trust model, he assigns a metric to that belief. His model uses a traditional decision making method based on utility derived from particular transactions and probabilities. However, in his model, the probabilities are not attached to the possible outcomes of a transaction. Instead the model is built on the assumption that the outcome is dependent upon whether the other party (agent) cooperates or defects. This allows the model to use trust measures to determine rational choice. (Josang, A., 1999)

In an earlier paper, Josang uses an anthropomorphic view of the environment in his definitions of trust. He categorized the mental process that a human makes when determining benevolent or malicious intent as "free will". Entities or agents possessing

free will are categorized as "passionate". As a result, Josang defined trust in passionate entities as "the belief that it will behave without malicious intent." (Josang, A., 1997)

Entities that do not have free will, such as algorithms, protocols, software, and hardware can also be trusted. These entities are called "rational". Rational entities do not have the capability of deciding malicious or benevolent intent. Therefore, trust in these entities must be determined by their security attributes. Josang defined trust in rational entities as "the belief that it will resist attack from malicious agents." (Josang, A., 1997) It is a matter of humans having volition, whereas machines do not have volition. However, Hal of '2001, A Space Odyssey' and Bender of 'Futurama' were ascribed anthropomorphic traits in which case they would be described by Josang as 'passionate.'

Others feel that the concepts of trust and security are very different. Security is an objective descriptor of whether an application, product or software meets a formal industry standard. Trust is dependent upon each link in the electronic transaction chain being secure. Each step from authentication with certificates, integrity through digital signatures to validation of an Internet site must be secure to ensure trust. (Hombeck, R., 1997) Security can also be viewed as a measure of how closely an information system implements the security policy of an organization or federation of organizations, or a community of users. In these definitions, security is the building block for trust.

C. MCCULLAGH'S DEFINITION OF TRUST

Adrian McCullagh defined trust in terms of electronic commerce. He believed that trust was a result of the interaction of four components of trust. An economic view is that market forces with the assistance of legislative mechanisms decide trust in the electronic commerce arena. However, McCullagh felt that technology trust and behavioral trust also influence trust in electronic commerce.

The first component of trust that McCullagh used to define an overall concept of trust is technology trust, which involves a classification of specific technology. Classification involves an analysis of the outputs of the computer system created by a particular process. Traditionally, technological trust resulted from the evaluation of the security mechanisms from which the system was composed. Again the evaluation

criteria set forth by the TCSEC, ITSEC, and Common Criteria were used to gauge the security of a system and assign it a security classification or sensitivity level. For trust to be established in an information system, the user must be able to assess the degree to which a system can carry out a security policy for a single or multiple levels of security classification (e.g., Trusted Solaris).

McCullagh posits that the user does not necessarily determine technical trust. Most users would not have the capability to properly assess the trust classification of a system. Instead, he hypothesizes that commercial market pressure dictates the proper level of technical trust necessary to conduct business. Banks and financial institutions are in the best position to dictate security levels. They have the ability to evaluate the risks and returns associated with various security levels. Credit card companies and banks have the most to lose from poor security, so it is in their best interest to establish minimum security requirements. In doing so they will dictate the technical risk a user faces while conducting electronic commerce. However the concept of technical trust has applications outside of commerce as well.

Behavioral trust is another component to the overall concept of trust. Every society has unique social norms. The concept of trust is viewed differently among various societies. Fukuyama in his paper "Trust – the Social Virtues in the Creation of Prosperity" broke societies into low trust societies and high trust societies.

The Chinese are very family oriented in that they do not generally trust people outside of their family. The French and Italians are also grouped into a low-trust society where family relations are essential for trust. One result of a low-trust society is that there is little distinction between ownership and management of a corporation. The owners do not trust others to manage their companies. Many companies in these societies are state owned.

The United States, United Kingdom, and Japan are categorized as high trust societies. In these societies, family relations are not critical for trust. In these societies, most companies have separated ownership from management. Shareholders have ownership in the company, but they trust a board of directors to properly manage the

company. Trust in the company is based on corporate information, past performance, and information about the directors. It is important to understand the environment to properly evaluate trust within diverse societies.

The third type of trust is product trust. This type of trust is achieved primarily through commercial marketing. This is analogous to brand loyalty. There are a number of factors that contribute to product trust. One of them is peer pressure. If everyone is using an IBM computer, then it is perceived as a good product, or it would not be so popular. Budweiser must be a good beer because it is the best seller in the United States. Familiarity is also important in establishing product trust. People trust a brand that they have experience with. They are more likely to try a brand that they have heard of in commercials rather than one they have never seen before. In the conduct of electronic commerce, people are more likely to use Internet sites and merchandise from companies with products that they trust. (McCullagh, A., 1998)

Another component of trust in electronic commerce is legal trust. He believes that without a legal framework, a sufficient level of trust by the parties to an electronic commerce transaction will not be attained. (McCullagh, A., 1998) There are many legal issues concerning electronic commerce that have not been addressed in the courts. One of the most significant issues is how to create legal contracts in electronic commerce. Components needed in electronic contracts such as witnesses, non-repudiation, and digital signatures have not been adequately addressed by legislative bodies and the courts. Another pressing concern is the treatment of evidence. For digital evidence to be presented in court, it must be authenticated with respect to its origin and the accuracy of storage, retrieval, and printing. Due to the fact that electronic media can be easily manipulated, additional steps must be taken to ensure the validity of the evidence. Another legal problem is personal jurisdiction. If a person has a pornographic Internet site in Texas, where it is legal, can he be arrested if an individual accessed his site from Arkansas, where those types of sites are illegal? There have also been a number of court cases concerning copyrights, intellectual property, and trademarks violations on the Internet. Although the courts have not been able to keep pace with the speed of

technology, they are steadily passing legislation covering technological issues such as the protection of intellectual property (e.g., music, software programs, movies, etc.) (Ford, W. and Baum, M., 1997)

McCullagh also hypothesizes that the degree to which each component of trust can be influenced depends on the circumstances in which trust is applied. There are three circumstances that he believes affect trust. These are normal circumstances, abnormal circumstances, and extreme circumstances. The types of circumstances define the level of risk associated with the trust. An example of an extreme circumstance would be an astronaut trusting an inertial guidance computer to bring him or her back to Earth. In normal circumstances, he or she may not trust a piece of gear that he or she did not program, but since the astronaut cannot build the entire system himself or herself, he or she has little choice but to trust it. An example of trust in an abnormal circumstance was the fact that some people did not fly on January 1, 2000 due to their lack of trust in aircraft systems that might be adversely affected by year 2000 (Y2K) software and hardware bugs.

Another definition of trust proposed by McCullagh is that absolute trust will come about when an entity totally relies upon an outcome after the entity has total control over the input and the process involved in producing the output. This is control trust. Obviously it is not possible to have total control, as one cannot control the environment from which inputs are received (i.e., outside the engineering design space), so the notion of belief trust was proposed. Belief trust evaluates external information and assigns probabilities to the accuracy, truthfulness, reliability, and completeness of that information. (McCullagh, A., 1998)

D. ESSIN'S DEFINITION OF TRUST

Daniel Essin defines trust as it relates to individuals in a socio-technical workplace, in particular, healthcare. Trust in the healthcare industry is extremely important, as the legal and ethical environment requires that special rights and authorizations be granted to healthcare professionals to perform their jobs. Tasks such as accessing personal information, surgical data, and psychiatric evaluations require a great

deal of trust from the patient. Essin believes that trust is a function of the following four elements: identity, reputation, capability, and stake. Different levels of trust are based upon the differences in these four elements. (Essin, D., 1997)

Before using a physician, his or her identity must be verified. This is to ensure that an imposter or charlatan is not providing the patient with medical care. Driver's licenses and employee badges can help to identify an individual. Hospitals routinely check social security numbers and national registries. Additionally, physicians are fingerprinted as part of their licensing.

It is difficult to assess an individual's capabilities with no prior knowledge of his or her abilities. Often capability is measured against education and training. Physicians must produce evidence that they have completed their training from accredited schools and programs. Additionally, most states require a physician to pass a state-sponsored test to ensure that a minimum competency is met before a physician can practice. Most organizations also closely supervise new members during a probation period to ensure that they have a practical application of the knowledge they are required to have to perform their duties and responsibilities.

Information about an individual's reputation provides an indication not only of past performance, but also provides a picture of the effect that person has had on others. Information about a person's reputation can be held by numerous sources. The easiest measure of an individual's reputation, although generally partial, is from letters of recommendations. However, phone calls to classmates, teachers, friends, and former employees also helps establish an individual's reputation. Awards, honors, and position in an organization are also an indication of an individual's performance.

Some organizations try to establish if an individual has a stake in the outcome of an organization. If an individual has a great deal of stock in a competing organization, that person is probably not going to be motivated to help his organization succeed. If employees share in corporate profits, or have stock options, they may work harder for the company because of their stake in the company. Unfortunately, a stake in outside interests can be very difficult to determine. Stake also applies to policies within an

organization. If a policy adds additional work without any benefits to that person, the policy may not be followed. If an individual can determine how a policy benefits that individual, then the policy will likely be successful because the individual has some stake in the policy.

The degrees or level of trust is dependent upon the weights assigned to the four elements. If the task to be performed is highly secretive, then the values associated with an individual's identity may be weighed more heavily than the other categories. If individuals are allowed a great deal of autonomy in their work place then either knowledge or stake may be more important. Each trust decision must be made in the context to which it is applied. (Essin, D., 1997)

E. ABDUL-RAHMAN AND HAILES' DEFINITION OF TRUST

Diego Gambetta in his article "Can We Trust Trust?" defined trust as a particular level of the subjective probability with which an agent will perform a particular action, both before one can monitor the action (or independently of his or her capacity of ever to be able to monitor it) and in a context in which that action affects our own actions. This definition raises three interesting issues. The first is that trust is subjective. The second is that trust is affected by actions that we cannot monitor. The third point is that trust depends on how an agent's actions will affect our own actions. (Abdul-Rahman, A. and Hailes, A., 1998, p. 49)

Abdul-Rahman and Hailes also describe the properties of truth in addition to their definition. The first property is that trust is not transitive. A chain of entities cannot generally transfer trust. If Bob trusts Alice, and Alice trusts Mary, it does not necessarily mean that Bob trusts Mary. Although trust is not transitive, "conditional transitivity" states the conditions in which transitivity may be allowed. Those conditions are as follows:

- Bob communicates his trust in Mary to Alice as a "recommendation".
- Alice trusts Bob as a recommender.
- Alice is allowed to make a judgement about the quality of Bob's recommendation.
- Trust is not absolute in that Alice may not trust Mary as much as Bob does.

Another property of trust is that it is not associative. If Bob trusts Alice concerning a particular subject, this does not mean that Alice trusts Bob equally concerning the same subject. If Bob trust Alice to be a competent driver, this does not mean that Alice trusts Bob to be a competent driver. (McCullagh, A., 1997)

Abdul-Rahman and Hailes hypothesize that it is important to understand how trust relationships work in order to model trust. They define two types of trust relationships and corresponding trust-relationship properties. A "direct trust-relationship" is one in which Alice trusts Bob. A "recommender-trust relationship" is the case in which Bob trusts Alice to give recommendations about another entities trustworthiness. This distinction is central to their trust model.

Trust relationships in their model have three properties. The first is that trust relationships must exist between exactly two entities. Their trust model measures direct and/or recommender trust relationships between each pair in a trust chain (e.g., A-B, then B-C and C-D). A direct measure of trust between A and D can not be measured without factoring in the trust relationships throughout the chain, unless a direct trust relationship exists between A and D. Trust relationships are non-symmetrical or (unidirectional). This means that if Alice trusts Bob, Bob does not also have to trust Alice at the same time. Trust relationships are also conditionally transitive. (Abdul-Rahman, A. and Hailes, A., 1998, p. 52)

F. GERCK'S DEFINITION OF TRUST

As we have seen, trust has been defined in numerous ways, with each definition being tailored to a specific model of the world. What is needed is a formal, abstract definition of trust that can be universally applied. This definition should define trust's properties, without citing any context or application. It should denote only behavior, not context. In this way both the environment and the observer are also abstract. We expect the abstract definition to contain trust's implicit conditions that can then be applied to explicit trust applications.

Claude Shannon, the father of information theory, had a similar difficulty trying to define the concept of information. He wanted to define information is such a way that

it could be widely applied to communication engineering, and have real-world application. He reasoned that information had nothing to do with knowledge or meaning. Information was simply waveforms, or electrical pulses that were sent from one point to another using a communication channel. If the receiving party already knew the information, then the value of the transmission was zero. So, information is that which the destination party does not expect, as measured by the uncertainty of the party as to what the message will be.

Ed Gerck defines trust in terms of its real-world application. He stated that for truth to be used in communications systems, it must be defined and understood as something that is potentially communicatable. Trust is subjective, but to communicate trust, it could not be defined in terms of something purely subjective, such as a feeling or a psychological belief. Trust cannot be expressed in terms of friendships, relationships, or loyalty. Additionally, for truth to be applied, it must be able to bridge different observers or instances, otherwise the communication of trust is isolated, and potentially not useful. An abstract definition of trust needs to be developed to communicate different subjective realizations of trust. Gerck stated "trust is that which is essential to a communication channel, but cannot be transferred from a source to a destination using that channel." (Gerck, E., 1998, p. 3)

The abstract definition can be applied to the real world better than an explicit definition that is dependent upon a particular set of environmental assumptions. With an abstract definition, different environmental assumptions are just different forms of the same abstract definition of trust, not different concepts. In other words, the abstract definition is the underlying proposition for all other definitions that one may derive from the concept of trust as it applies to the observer and environment. To properly apply an abstract definition, explicit definitions need an explicit stance and an explicit observer. This allows one to evaluate different trust models in relation to one guiding abstract definition.

Gerck gives an example of the definition by explaining the need for trust in communication between a lion and a lamb. His example also shows how trust and power

are related. When a lion communicates with the lamb, the lion only needs the information that is communicated in the channel itself. The lamb on the other hand needs to know if the lion is hungry (does the lamb trust the lion). This is information that cannot be passed along the communication channel. If the information that the lion is hungry is passed along the communication channel, then the lamb did not know that data previously. By that time the lamb has been eaten. Additionally, how can the lamb evaluate the accuracy of the information received from the lion, if that information was passed along the same communication channel?

This example also illustrates how a balance of power can affect the use of trust. The lion can dominate the actions of the lamb, so the lamb's actions are immaterial. As such, no trust is needed from the lamb, as the lion dominates its actions. The lamb, conversely, needs trust on the lion's behavior. Since it cannot influence the lion's behavior it must know with a high level of reliance what the lion's actions will be.

The example also illustrates another point. If trust is based on what you know, then it is also essential that you know how to act upon that knowledge. The lamb must not only know that the lion is hungry, but it must also know what action to take based on that knowledge. One cannot use prior knowledge about an event, unless one also knows what actions to take in response to the event.

Gerck demonstrated over 30 examples of explicit definitions of trust. He derived these using the abstract definition of trust and applied other concepts and views. He combined Shannon's abstract definition of information and his abstract definition of trust to derive his first explicit definition.

Shannon's definition of trust is essentially that "information is what you do not expect." Gerck's definition of trust is "trust is what you know." By combining these two concepts, an explicit definition of trust can be derived. So in this context, "trust is qualified reliance on received information." (Gerck, E., 1998, p. 5) It is now possible to use this explicit definition with other stances to derive more explicit definitions. In this way, trust in almost any situation can be defined in a way that represents the underlying abstract definition of trust. If all trust models use a definition of trust derived from the

same abstract definition of trust, then interoperability is greatly increased as similar metrics can be applied.

By adding a subject, an observer, and a metric, another explicit definition can be derived from the first explicit definition that "trust is qualified reliance on received information." The new explicit definition can now be stated as "trust is that which an observer knows about an entity and can rely on to some extent."

Gerck discussed the concept of quasi-zero variance as a measure of what the observer knows about an entity and can rely upon. Quasi-zero is a positive number that is approximately zero. The closeness to zero is a subjective measurement that is determined by the observer. So quasi-zero variance is essentially the same as 100 percent reliability.

Models, or definitions of trust must be defined in relation to some point in time. Trust before an event can be significantly different after that event. So, trust must include a statement representing the time frame in which it is applied.

Using the concepts of variance and time, a new explicit definition can be derived. The new explicit definition is "trust is that which an observer has estimated with quasi-zero variance at time T, about an entity's unsupervised behavior on matters of X". (Gerck, E., 1998, p. 6) "Matters of x" refers to a specific occurrence, event, or subject. The term 'estimated' is not just a probability. It also takes into account other notions such as inference, deduction, constraints, and past knowledge. This is an important concept because it means that another metric must be used to measure the term 'estimated.'

Because probability is not the primary metric used to describe the term 'estimated', the term 'justification' is used as a measurement of certainty. Probability and deductive logic are elements used to determine a level of justification for a subject matter, but there are other subjective elements that can be used instead or in conjunction.

"Best justification" is the justification level that equates to 100 percent certainty. The user determines how that level of certainty is achieved. There are various types of reliance that can be used to evaluate the justification. "Justified reliance" is a subjective

belief based upon an examination of the facts presented. "Reasonable reliance" is what a reasonable man might do. "Actual reliance" is what the truster actually relied upon without any consideration as to why. "Random reliance" is what a random process might choose given all of the possibilities. "Authorized reliance" is the term that the truster accepted from an outside source, such as a company. "Process reliance" is the reliance on a technology to automatically make a decision. Although it is a conceptual metric, all meanings of trust can be captured by "best justification" because it is the user that defines "best" and "justification" as it applies to "matters of x". (Gerck, E., 1998)

Gerck measures degrees of trust not on the estimator's quasi-zero variance, but on the size of "matters of x". In his model, quasi-zero variance never changes. Rather the subject of trust (as defined by matters of x) are increased to reflect a high degree of trust, or lowered to reflect a reduction in the amount of trust. This is similar to using Z values to determine the area under a curve that equates to a specific standard deviation variance. This means that if the truster has no trust on the observed entity, then "x" equates to the empty set. Accuracy is measured by the extent to which "matters of x" provides high reliability. Thus, as a trustee increases his degree of trust about an entity, the estimator represents a larger set of "matters of x" that have quasi-zero variance.

This method of describing degrees of trust is superior to terms used in other trust models. Terms such as 'partial trust', 'fully trusted', and 'marginal trust' describe different levels of trust. However, these terms are vague, ambiguous and are difficult to define in quantitative and qualitative terms. Additionally, it is difficult to describe small or atomic changes in trust using these terms. Descriptors such as good, bad, marginal, complete, or maximum should not be used in conjunction with the term 'trust', because they are ill defined, and subject to misinterpretation.

The terms unqualified trust and qualified trust also present ambiguity. Unqualified trust is represented by "Bob trusts Alice." This statement is difficult to interpret because it is not clear as to the subject with which Bob trusts Alice. Does this mean that Bob trusts Alice on all "matters of x"? If so, then x is the universal set. Given that statement, then the statement Bob does not trust Alice indicates that x is zero or the

null set. A better and more formal statement of trust can be obtained by defining "matters of x" in the trust proposition (quasi-zero variance).

Gerck prefers to measure trust in quantitative terms. Qualitative terms provide for a weak ordering, but one does not know if a trust value is closer to being "partially trusted" or "marginally trusted" when using such terms. Numerical values in conjunction with qualitative terms present to the user multiple views of the level of trust that can be adapted to varying situations. One must, however, question the subjectivity involved with assigning the numerical values used to measure trust. Gerck's method of describing trust assumes that there is a standard systematic method of determining the size of "measures of x." Additionally, there can be varying interpretations on what constitutes a particular "measure of x."

Trust and belief are often confused. Trust is not the same as belief, but it can be expressed in terms of belief. Gerck states "belief is the probability that the evidence supports the claim." (Gerck, E., 1998, p. 10) Thus belief can be used to gauge reliance on a trust point (matter of x). Belief can be used to verify if the trust point represents an entity's actual behavior in light of the evidence. Combining the definitions for belief and trust, a new explicit definition of trust can be defined by "trust is received information, which has a degree of belief that is acceptable to an observer." (Gerck, E., 1998, p. 10)

Trust and risk are interrelated. Indeed, risk is a component of trust. If there is no perceived risk, then there is trust. If there is certain risk, then there is no trust. Both concepts are subjective and are perceived by an individual. However, the lack of risk does not necessarily mean there is quasi-zero variance, because trust is often based on more aspects than just perceived risk.

We have stated that trust is subjective, but to this point we have not defined what subjective means in relation to trust. Gerck states that the term 'subjective' means that one needs to take a personal instance in order to evaluate an object. For example, beauty is in the eye of the beholder. Another descriptor is 'intersubjective.' To be intersubjective, an instance can yield different results for objects of the same class or type. This means that the object must be capable of being instantiated differently. A

'football play' is intersubjective because the play is a particular instant from the class of all plays available for that team at that time. Based on these definitions, trust is subjective, as it is an abstract object that cannot be instantiated. However, trust in an object such as a certificate, which can be instantiated, and intersubjective. So trust can have intersubjective dependencies.

The abstract definition leads to some postulating statements concerning trust. The first is that truth is subjective, so it depends upon the observer. There is no absolute trust as it pertains to everyone, because everyone views things differently. Another is that trust only exists as "self-trust". Self-trust is what an individual knows. It is knowledge that the individual has about himself or herself and the external environment. It does not however include knowledge that the individual does not know he or she has. Based upon the first two statements, it is reasonable to assume that two different observers cannot equally trust any received information (except by coincidence), as the interpretation of trust can differ from one individual to another and the fact that trust is based on self-trust. Finally, a person cannot convey trust to another entity when using the same communication channel. This would violate the abstract definition that states that trust is essential to communication, but cannot be conveyed using that same channel.

According to the abstract definition of trust, trust cannot be communicated. Trust is what you know. If information is transferred by modulating a carrier wave, the carrier transfers no information, while the modulating wave communicates all of the information. Using this analogy, trust is not the modulating wave; it is the carrier itself. So trust is basically a carrier of information. If information is communicated along a channel, it is either acted upon or not, depending on the underlying trust.

Information is not enough to support decision-making. Factors such as the accuracy, timeliness, and completeness of the information also are essential for decision-making. Trust is the subjective evaluation of these factors that permits an individual to determine his or her level of confidence in the information.

As one can see, there are numerous definitions of trust. Because trust is a subjective notion, the definitions apply many different variables and concepts. The

relevance of the viewer and the environment in which trust is being defined is important for evaluating the various trust models. Unfortunately, there are models and protocols in use today that deal with trust, but they do not contain an explicit or concise definition of the concept of trust that they are modeling. Once again we ask the question, "how can you model something that you cannot define?"

IV. TRUST MODELS

A. MODELING TRUST

Trust models were developed in an attempt to automate and make explicit the logic, variables, and thought processes that a human performs when making a trust decision. Trust models dissect trust into element parts, categorize trust, and assign metrics to trust. The designers of trust models also try to develop a system to communicate a notion of trust from one entity to another. Since trust is a subjective belief, the models assign a metric to a belief variable that will have value when evaluating trust.

As trust is a subjective notion, one can expect that the methodology used to assess trust can differ significantly from one model to the next. Models also differ in the metrics and variables used in trust decision making. We have included a discussion of the most popular and comprehensive trust models along with a discussion on possible weaknesses.

B. REITER AND STUBBLEBINE'S METRIC PRINCIPLES

In trust models, metrics are used to measure the trust. These metrics can be formulas, algorithms, or a methodology. Michael Reiter and Stuart Stubblebine developed a set of 8 principles that they felt a good authentication metric must meet. The principles center around three topics. The first is whether the output of the metrics is meaningful. The second measures the extent to which the metric can be manipulated by malicious forces. The last topic is concerned with the ability of the metric to be applied to practical matters. (Reiter, M. and Stubblebine, S., 1999)

The first principle is that models should not assume a public key to owner binding. It is easy to determine that a key has signed a document by decrypting it with its corresponding public key. It is more difficult to determine the actual owner of the key. The user should use a metric to measure the trust that is placed in the key to owner binding instead of the user making that assumption before applying the metric.

The second principle states that the meaning of the model's parameters should be clearly defined. This applies especially to probabilities and trust values used in models. Unless a model's parameters are well defined, interpretations in the metric values can be significantly different. This is especially important when relying upon metric values from another entity.

The third principle is that a metric should take into account as much information as possible before a decision is generated. This principle is intended to measure the ability of the model to incorporate any additional information. If the model is too narrow in scope, it may not consider other factors that may affect trust.

The fourth principle states that a metric should consult the user for any authentication related decisions that cannot be accurately automated. Unless the metric can arrive at an authentication decision through well-defined, intuitive heuristics, the user should be consulted. A metric should not use default values without the user's knowledge of what those values are.

The fifth principle is that the output of the metric should be intuitive. The output should be unambiguous, so the user can determine if the metric generated an output that was expected. Unless an output is intuitive, a user cannot decide if a metric is effective to use for a given application. If the output is ambiguous it is not useful.

The sixth principle is that a metric should be designed to resist malicious action. If the metric is susceptible to malicious behavior, the user should be informed. If a model can be significantly manipulated by one incorrect value, whether it is incorrect due to a malicious act or error, the user should be aware. This is especially important if those values are generated by other entities.

The seventh and most obvious principle is that the metric must be computationally efficient. As models get very large, their computations can grow exponentially.

The last principle is that the metric's output given partial information should be informative. The metric output given partial information should lead to a conclusion about a metric output. It does not have to be as accurate as output given full information,

but it should give general information. In a large-scale system, it may not be possible to get all of the information required, so the metric must act on the information it is given and produce a meaningful result. If a metric can determine an upper or lower bound for an output given partial information, then it is providing useful information. (Reiter, M. and Stubblebine, S., 1999)

C. DANIEL ESSIN'S TRUST MODEL

Daniel Essin modeled trust as a method to identify the elements that determine a trust decision. By understanding how various elements affect trust individually and synergistically, policies that rely on trust can be developed more efficiently. Essin believes that trust is a function of identity, reputation, capability, and stake. All of these elements are represented in his trust model. Essin's model also attempts to account for social and cultural factors that may affect trust.

In his model, the activities that an entity is engaging in is represented in his model by an 'a'. To be included in the model an activity must be subject to policy considerations. Activities are subject to policy considerations if they consume resources, have the potential to alter the state of an organization; or they must occur to protect or preserve resources. An entity, consisting of a subject or object, that is being evaluated is represented by an 'e'. The subjects, objects, property, or resources that are affected by an activity are represented by a 'd'. A 'c' represents the context for an activity, or the association that an entity has with a particular activity.

In Essin's model, valuation is a quantitative or qualitative measurement of the resources or assets affected by an activity given the context. The letter 'v' represents valuation.

Benefits assess the cost or utility associated with performing a particular activity. Benefits or 'B' measures the valuation associated with the activity and the entity. Stake, represented as $S_{(e,c,d)}$, measures the degree to which an entity has a vested interest in the outcome of an activity. It evaluates the entity, the context, and the results the activity has on various resources or assets.

Knowledge is the demonstrable expertise that an entity possesses about the activity. It also represents the authority the individual has to take action. Knowledge is represented in the model as $K_{(e,a,d)}$. Reputation, represented as $R_{(e)}$, attempts to model the effect that the entity has on others. It also takes into account historical information that may be used to assess the current situation. Identity is a measure of the certainty that the true identity of the entity is known. It is represented as $I_{(e)}$.

The general form of Essin's trust model is $T = f(B, S_{(e,c,d)}, K_{(e,a,d)}, R_{(e)}, I_{(e)})$, where $B = {}_{b(a,v,e)}$. Trust is a function of benefits, stake, capability or knowledge, reputation and identity. (Essin, D., 1999)

1. Weaknesses in Essin's Trust Model

Essin's trust model was developed for theoretical purposes vice practical application. His model does not specify a means of assigning values to his metrics. He also did not formulate an algorithm to compute a level of trust. His model is worthy of discussion in that it introduces interesting variables for consideration. He has done a good job trying to identify variables that a human uses to make a trust decision.

One variable that Essin has not used is uncertainty. Trust is a function of the information that you know, as well as information that you cannot ascertain adequately. His model assumes that the values that are assigned in the model account for uncertainty, but how is that information conveyed to another person. If the variable reputation is assigned a numerical value between 1 and 0, does a .5 indicate an average value for reputation, or total uncertainty?

D. PRETTY GOOD PRIVACY (PGP)

Pretty Good Privacy (PGP) is a public-key cryptography program that was developed primarily by Phil Zimmerman in 1991. It is used primarily to provide confidentiality and authentication for electronic mail. It has gained worldwide popularity due to its strong cryptographic algorithms, the fact that it is available at no cost (open source), and its adaptability to the Internet. PGP uses a unique trust model to determine the trustworthiness of the certificates exchanged among users.

In the PGP system, each user generates public and private keys. As there is no central repository to store public keys, each user must exchange public keys with individuals he wishes to correspond with. To manage and store the public keys of other users, PGP devised a system known as key rings. Each user has a private and a public key ring.

The private key ring contains the public and private key pair of the owner. It consists of a timestamp, key identification, public key, private key, and a user identification. The timestamp indicates the date and time that the key pair was generated. The key identification is the least significant 64 bits of the public key. If a user has more than one key, the key identification is used to distinguish them. The public and private key fields contain the key pair that was generated by the algorithm. The user identification is usually the user's e-mail address, but the user can put a name in this field as well.

The public key ring is used to store the public keys of other users. The public key ring consists of a timestamp, key identification, public key, owner trust, user identification, key legitimacy, signatures, and signature trust. The timestamp, key identification, user identification, and public key are the same fields as those used in the private key ring, except a public key can have a number of user identifications associated with it. The other fields are used to determine a level of trust in a public key.

For Bob to send an encrypted e-mail to Alice utilizing PGP, Bob must have Alice's public key in his key ring. Bob can get Alice's public key from a trusted individual, a bulletin board, or directly from Alice. Depending upon the method of obtaining the key, Bob must determine how much he trusts that the public key belongs to Alice. The key legitimacy field, signature trust field, and the owner trust field are used to indicate the extent that PGP will trust the public key.

If Tom knows Alice's public key, and Bob knows and trusts Tom's public key, then Tom can send Alice's public key to Bob. When he sends the key he will create a hash of the key and encrypt it with his private key. In PGP this is called signing the key.

Multiple signatures can be collected for the same public key. These signatures are stored in the public key ring.

Every entry in the public key ring has an owner trust field. In this field the owner is the individual that signed the public key certificate. This field represents the trust that the owner of the public key ring has in an owner to recommend a public key certificate. This field would measure the extent that Bob trusted Tom to recommend or sign Alice's public key certificate. The following levels of trust can be used in the owner trust field: undefined trust, unknown user, usually not trusted to sign other keys, usually trusted to sign other keys, always trusted to sign other keys, and ultimate trust where the person owning the public key ring signed the certificate. The owner of the public key ring assigns these levels of trust.

The public key ring also has a field called signature trust. The signature trust field represents the same trust values that are assigned to the owner of the signature. The entries in this field are used by PGP to make a final determination on the degree to which an individual public key certificate is trusted.

Each entry in a public key ring has a key legitimacy field. This field is computed by PGP. It determines PGP's level of trust in the public key to user identification binding. The following levels of trust can be assigned in the key legitimacy field: unknown or undefined trust, key ownership is not trusted, marginal trust in key ownership, and complete trust in key ownership. (Stallings, 1999)

The entries in the key legitimacy field are calculated using the values assigned in the signature trust field. PGP computes a final trust value using a weighted sum of the signature trust values. Signatures that are always trusted are given a weight of 1/X. Signatures that are usually trusted are given a weight of 1/Y. The user defines the variables X and Y. When the sum of the values reaches 1, the key is considered trustworthy, and the key legitimacy field is set to complete trust in key ownership. In many PGP settings 1 always trusted signature or 2 usually trusted signatures are required for complete trust.

Although the key legitimacy field indicates complete trust, this trust only applies to the owner identification to public key binding. It does not represent trust in the owner of the certificate to recommend other public keys. That trust is represented in the owner trust field.

When Bob enters a new public key on his key ring, PGP must assess the level of trust associated with the new public key. PGP will ask Bob to assign a trust value to the individual signing the public key. The values to be assigned will be entered in the owner trust field.

When the key is entered into the public key ring, one or more signatures may be attached to the certificate. If the signer of a certificate had already been assigned a trust value in the owner trust field, that value will be assigned to the signature trust field. If the owner is not known previously, the signature trust field will reflect an unknown signature.

PGP then uses the weighted sum of the signature trust field to calculate the key legitimacy field. This final value is the extent to which PGP trusts the certificate to owner identification binding.

As users trade public keys, a chain of trust is created. This is a simple form of PKI because each user is his own certificate authority, with total control over how trust values are assigned.

1. Weaknesses of Pretty Good Privacy (PGP)

PGP works well in a small domain (i.e., among friends or in a small business), however, its uses in e-commerce and other applications that require strong authentication are limited. PGP relies on an e-mail address for authentication. Since an e-mail address cannot provide a key to owner identification binding, PGP's use is limited.

Another major weakness of PGP is that it does not have a system to revoke certificates. If a certificate is compromised, users must rely on others within the chain of trust to inform them. It is difficult to validate if a certificate has in fact been revoked, because there is no third party verification. The information could have been forged or tampered.

Changing a key can also be very difficult. If Alice changes her key, she must let everyone in her chain of trust know that the key has been changed. However, how can the recipients of Alice's message know if Alice sent the message, or whether her key has been compromised, and someone else is sending the message? Alice must communicate with her chain of trust by means of an out of band method. Additionally, if Alice's key has been compromised, she must exchange a new key with her chain of trust. How will she pass the new key to her friend Tom? She cannot use the old key, because it has been compromised. Tom will have no way of confirming if a new key for Alice was actually sent by Alice, unless she contacts him using an out of band channel.

The PGP trust model forces the user to make some important decisions outside of the model. The method in which a user determines who is trusted and to what extent is determined outside of the model. The model does not contain any information that would assist the user in making that decision. Signature input from others helps to determine key legitimacy, but not whether the owner of the certificate is trusted to recommend others. Only the owner can assign those values. Additionally the model does not list any characteristics or traits that a trusted recommender or a trusted key hold.

PGP also violates the fourth principle specified by Reiter and Stubblebine. A metric should consult the user on any relevant decision that cannot be accurately automated. PGP does not take into account the fact that one person may have more than one key. PGP declares a key legitimate if one fully trusted key, or two partially trusted keys sign the certificate. Unfortunately, if a partially trusted individual has two keys, both keys can be used to sign a certificate and make it a legitimate key. PGP will believe that two separate individuals, both of whom were partially trusted, signed the certificate. (Reiter, M. and Stubblebine, S., 1999)

PGP certificates do not allow attributes. As a result, the trust that is transferred in a certificate is universal trust. PGP does not allow certificates that can identify a 'matter of x' for which trust is assigned.

Another weakness of PGP is that users can sign a certificate based on information from third parties. A receives a certificate from D that is signed by partially trusted

individuals B and C. If user A specified that two partially trusted signatures were required to accept a key as legitimate, then D's certificate would be legitimate. This assumes that B and C had first-hand knowledge of D. If both B and C received their recommendation on D from individual Z, then A is in fact declaring the key legitimate based on the recommendation from Z. A does not even know Z. (Josang, A., 1999)

E. ABDUL-RAHMAN AND HAILES TRUST MODEL

Alfarez Abdul-Rahman and Steven Hailes use conditional transitivity to model trust. An example of transitivity is if Bob trusts Alice and Alice trusts Oscar, then Bob trusts Oscar. This is usually not the case. Bob will probably not trust Oscar as much as he trusts Alice, because he knows Alice personally. Transitivity does not apply well to trust. As a result, Abdul-Rahman and Hailes developed the concept of conditional transitivity. They believe that under conditional transitivity, trust can be transitive.

Conditional transitivity adds another variable called recommender trust. This is the trust that one individual has in another individual to properly recommend another entity. The authors believe that there are four conditions that must be met to achieve conditional transitivity. The first condition is that Bob explicitly communicates his trust in Tom to Alice in the form of a 'recommendation.' The second condition is that recommendation trust exists, such that Alice has trust in Bob as a recommender. The third condition is that Alice is free to determine the level of trust that she is willing to assign to Bob as a recommender. The final condition is that trust is not absolute. Although Bob recommended Tom to Alice, Alice does not have to trust Tom as much as Bob does. (Adbul-Rahman, A. and Hailes, S., 1997)

In their trust model, the authors use explicit trust statements. They use trust categories to define what is actually being trusted. The statement 'Bob trusts Suzie,' means that Bob trust Suzie in all things. The category of trust is universal. To lessen ambiguity in trust statements, trust categories are used that explicitly state what is being trusted. This is similar to Gerck's notion of 'matters of x.'

The authors believe that for a trust relationship to exist, the following properties must be met: trust is always between exactly two entities, trust in unidirectional, or non-

symmetrical, and trust is conditionally transitive. Unidirectional trust refers to the fact that Bob may trust Alice, but that does not mean that Alice trusts Bob to the same extent.

There are two types of trust relationships used in their model. The first is direct trust, which means that Alice trusts Bob in 'matters of x.' The second is a recommender trust relationship. This is a relationship in which Alice trusts Bob to give recommendations about another entity. This means that an entity can be a recommender, or a requester of a recommendation.

In addition to trust categories, Adbul-Rahman and Hailes use trust values in their trust model. For direct trust, the scale is between negative one and four. Four represents complete trust and negative one represents complete untrustworthiness. A zero represents ignorance. The trust values for recommender trust are also between negative one and four.

A request for information from a recommender is called a recommendation request message (RRQ). The requestor will receive a recommendation in response to the RRQ. The RRQ has the following format: RRQ = (Requestor_ID, Request_ID, Target_ID, Categories, RequestorPKC, Get PKC, Expiry). Categories are the set of trust categories that the requestor is interested in. RequestorPKC is the public key of the requestor that can be used to encrypt a recommendation, if the recommender feels that the information is sensitive. The GetPKC is a Boolean flag, which when set to true, means that the requestor would like the target's public key certificate to use in further communication. The Expiry field contains the date that the RRQ is no longer valid, ensuring that requests do not travel around the Internet indefinitely.

The recommendation that is generated in response to a request, has the following format: Recommendation = (Requestor_ID, Request_ID, Rec_Path, [Sequence of Recommendation_Set, TargetPKC | Null]. The Rec_Path is the trust path; it contains an ordered sequence of recommender_IDs. The Recommendation_Set consists of the set of Recommendation_Slips. A Recommendation_Slip contains the information that the requestor desires. It has the following format: Recommendation_Slip = (Target_ID, Category_Name, Trust_Value, Expiry). In this case Expiry is the validity period of the

recommendation. After this date, the recommendation should not be used. If no recommendations are found that satisfy the request, a null value is entered in the Recommendation_Set.

Bob owns a company that installs networks. He is looking to hire a person that is very familiar with Cisco routers, so he asks his friend Alice. Alice knows that Susan has a friend that works with routers, so she asks Susan. Susan has first-hand knowledge of Jack's experience with Cisco routers. In this scenario, Bob is the requestor, Alice and Susan are the recommenders and Jack is the target. The trust category is Jack's reputation in working with Cisco routers, so we would represent the trust category as "Router_Programmer".

Bob's request to Alice would have the following format: Bob, rrqA01, Jack, [Router_Programmer], T, 20001001. Alice's request to Susan would be: Alice, rrqB01, Jack, [Router_Programmer], T, 20001001. Susan's response to Alice would have the following format: Alice, rrqB01, [Susan], (Jack, Router_Programmer, 3, 20000901), PK_{Jack}. Alice's response back to Bob would be: Bob, rrqA01, [Susan, Alice], (Jack, Router_Programmer, 3, 20000901), PK_{Jack}.

This scenario assumes that Bob trusts Alice as a recommender, and she similarly trusts Susan as a recommender. Susan has direct trust with Jack and is able to convey that trust back to Bob. If Bob was unsure of the trust path, he can check the Rec_Path to obtain a listing of everyone that helped him obtain his information.

Since Bob does not know Susan directly, he can ask Alice about Susan's trustworthiness as a recommender. The request would be: Bob, rrqA02, Susan, [Rec_ Router_Programmer], F, 20000901. Rec_Router_Programmer is the trust category for recommending a router programmer. Alice's recommendation would be as follows: Bob, rrqA02, [Alice], (Susan, Rec_Router_Programmer, 4, 20001001), Null.

To update a recommendation, the recommender sends a refresh message. The refresh has the following format: Rec_Path, Recommendation_Set. If Susan found out that Jack had forged his Cisco certifications, she could send the following refresh

message to Alice: [Susan], (Jack, Router_Programmer, -1, 20000901). Alice would then forward the information to Bob by appending her name to the Rec_Path.

If a recommender wanted to revoke a recommendation he or she would send a refresh message with the trust value set to 0. When other entities receive the revoke message they will no longer use the recommendation.

If a requestor needs to tune his piano, but cannot gather any information on who can tune his piano, he can send a RRQ to all of his trusted agents to request the information. A question mark in the Target_ID field lets everyone know that the requestor is not looking for any one individual in particular. The RRQ would have the following format: Bob, rrqA01, ?, [Tune_Piano], F, 20001001. He may get a response from Margo that looks like this: Bob, rrqA01, [Margo], (Liberache, Tune_Piano, 2, 20000901), (Joshua, Tune_Piano, 3, 20000901), Null.

To determine the trust value for a single trust path, the following formula is used: $tv_p(T) = tv(R1)/4 \text{ X } tv(R2)/4 \text{ X } \ldots tv(Rn)/4 \text{ X } rtv(T)$. The symbol '$tv_p(T)$' is the trust value of target T obtained through the trust path p. The symbol 'tv(Ri)' is the recommender trust values of the recommenders in the trust path. This measures the extent that the requestor trusts them to give recommendations. The final symbol 'rtv(T)' is the recommended trust value that was in the recommendation.

If multiple paths are used to obtain a trust value on the same target and trust category, the final trust value is a weighted average of the trust results. The formula is as follows: $tv(T) = \text{Average}(tv1(T), tv2(T) \ldots tv_p(T))$. If one path had a trust value of 1 and another had a trust value of 2, the total trust is computed as the average of the two values, or 1.5.

1. Weaknesses in Abdul-Rahman and Hailes Trust Model

The algorithm used in the model has flaws. The algorithm does not explain how to deal with the values for distrust (negative one) and ignorance (zero). Either of these values will significantly affect the final trust value. It will either be a negative number or a zero. This is a violation of Reiter and Stubblebine's 6th principle, that a metric should be designed to be resistant to malicious entities. If any value is changed to reflect distrust

or ignorance, the corresponding change in the final trust value could be large. In addition, if two distrusted people were in a path, multiplying (negative 1/4) by (negative 1/4) will give a positive result of 1/16, which is the same a two marginally trusted people.

This model also does not adequately deal with the concept of recommender trust. Bob has assigned a recommender trust of 2 to Alice. Bob asks Alice about the trustworthiness of Susan as a recommender. If Alice (who has a recommender trust value of 2) returns a recommender trust value of 4, does Bob then assign a recommender trust value of 4 to Susan, or does he assign a lesser value? The model does not address this problem.

The authors make the assumption that authentication between agents have been established. Authentication was undoubtedly omitted to concentrate on the core of their paper, which was a method to convey trust. However, authentication adds additional complexity to their model, and this topic was not specifically mentioned as being omitted from the model.

F. REITER AND STUBBLEBINE'S TRUST MODEL

The Reiter and Stubblebine trust model is designed to provide trust in a name or attribute-to-key binding. It is based on the concept of one entity bonding another. This is similar to the commercial practice of bonding where an organization will insure the actions of another organization. The nodes in the model are public keys (i.e., K1, K2). An edge exists between the nodes if the user is in possession of a certificate that assigns attributes to K2, but was signed by K1.

In addition to attributes, each edge contains a label that specifies the amount of money that K1 insures for the attributes of the certificate and the behavior of K2. K1 is liable for the amount specified on the edge if the attributes assigned to K2 are incorrect. If the private key corresponding to K2 is compromised and used maliciously, then K1 is liable. The insurance label on the edge must be obtained from K1 and is also carried in K2's certificate that represents the edge from K1 to K2.

This model is not concerned with establishing a key-to-owner binding for K1, because the entity that certified K1 is held liable for any misrepresentation of K1. Thus a trust path can be set up between a trusted source key and a target key.

Once a path of liable edges have been established from the trusted source key to the target key, the metric computes all possible paths and returns the minimum amount of money that a user can expect to recover. This is the minimum amount to insure a key to owner binding.

The authors envision that insurance agencies will perform the task of trusted key. They will in effect act as a certification agency (CA). The insurance organizations will collect revenue from two types of users. The first user is the individual who is willing to pay the insurance organization to create a certificate that will bind the owner's attributes and name to a key pair. The certificate will bind agent A to key K_A, and will be insured by insurance organization C.

The second type of customer pays the insurance organization for the use of a certificate to authenticate a key. To obtain the certificate that binds A to K_A, and is signed by K_C, the customer must pay the insurance organization. If an individual wants a certificate to follow a trust chain, he must pay the insurance organization that signed the certificate. He would then be using that insurance organization as the trusted source key. Users could also purchase certificates from multiple insurance organizations to create a number of paths from multiple trusted keys to the target key. This would reduce the reliance on any one organization. (Reiter, M. and Stubblebine, S., 1999)

1. Weaknesses in the Reiter and Stubblebine Model

This model assumes that the user of the model is able to assign a value to a particular transaction. If the user receives a certificate that is malicious, can the user place a value on the damage that the malicious certificate can cause? The model assumes that the user can perform a cost benefit analysis and determine if the insurance that the path offers will offset the risk of the transaction. It can be difficult to assign values to attributes such as reputation, good will, inconvenience and trust. It is also difficult to

estimate the total man-hours, which may be required to correct a situation that a malicious certificate caused.

The model makes the assumption that the user has complete trust in the certificate authority (in this case it is the insurance company). There is no means for a user to determine if insurance company A did in fact insure certificate B. For a name-to-key binding to be effective, trust must be measured in three areas. The first is the trust in the name-to-key binding, which in this model is represented as a monetary value. The second is the certifying authorities trust that the company or individual are who they say they are. In the model this is represented by the insurance companies willingness to insure the company or individual. However, the willingness to insure could be argued because the companies are charging the individuals for this service. As long as their costs are being covered, do the certifying agents really trust the entities they certify? The last area of trust is not covered in this model. That is the trust that the user has that the certificate authority (insurance company) signing the certificate is valid.

The author does not mention how the insurance companies will determine insurance values. Do they assign the same insurance values to all individuals, or is there some type of screening process? How do you assign insurance to a company? Is a corporation more trustworthy than a partnership or single proprietor? Does the individual request a specific level of insurance and pay the appropriate premium? The method of assigning insurance can also influence a person using the path. If a company insures the least trustworthy certificates for 30 dollars, a user may not want to use that certificate, even if the 30 dollars would cover the cost of the transaction. Psychologically, there are many reasons that people do not like dealing with the lowest level of certificate (e.g., risk aversion). They may use another path, even if that path will provide a lesser amount of insurance coverage, if they believe that the trustworthiness of the certificates are higher.

Conceptually, this is a very strong model, however there are a number of practical, real world issues that would have to be resolved before the model could be implemented. The authors acknowledge that there are numerous practical issues that this model does not address. Some of the practical issues to consider are as follows: method

of recovering money from insurance companies, identifying liable parties, and payment of insurance premiums. (Reiter, M. and Stubblebine, S., 1999) Other issues are the costs to small businesses, industries acceptance of the model, and regulatory bodies to govern the insurance companies. (Ford, W. 1997)

The model does not account for any aspect of time. Presumably the certificates have expiration dates assigned to them and the insurance will extend for the time life of the certificate, however this aspect was not mentioned.

G. AUDUN JOSANG'S TRUST MODEL

Audun Josang's trust model was developed for use in the authentication of public keys. In an open environment such as the Internet, certificates alone cannot validate authenticity. The trust in the binding of a certificate key and its owner is essential in providing a level of legal culpability (i.e., digital certificates and non-repudiation). The certifying agents that created the certificate must also be assessed for trustworthiness. Do they properly check identification before issuing a certificate? It is important to note that a key's authenticity can be validated by its corresponding public or private key. The certificate that holds the key is what needs to be validated.

Josang defined trust as a subjective measure. He defined trust in a system as the belief that it will resist malicious attacks. Trust in humans was defined as the belief that he or she will cooperate and not defect. (Josang, A., 1999) In his model, he assumes that the outcome of a transaction depends on whether an agent defects or cooperates. Thus, probabilities are not assigned to possible outcomes. Instead, trust measures are used as a decision mechanism.

Josang's trust model is based on the belief about the truth of statements. The statements must be crisp (i.e., they are either true or false). Whenever the truth of a statement is assessed, it is always done by an individual, and thus is a subjective determination. The belief in a statement is purely binary. Humans do not have perfect knowledge, so it is impossible to know with certainty, whether a statement is true or false. We can only have "opinions" about the veracity of a statement. These opinions represent degrees of belief, disbelief, and uncertainty. Josang expresses "opinions"

mathematically as $b + d + u = 1$, $b, d, u \in [0,1]$, where b, d, and u represent belief, disbelief, and uncertainty.

Atomicity is also an element of an opinion. Relative atomicity, denoted by the letter 'a', is used to describe the atomicity of a sub-state in relation to a full state space. This concept can be further explained with an example. There is a bin in which the ratio of black to red balls is unknown. With no knowledge of the ratio of red and black balls in the bin, most people would agree that the probability of selecting a red versus a black ball were equal, so the probability of selecting a red ball would also be 0.5. In this situation, there is total uncertainty, so simple probabilities do not account for the level of trust in selecting a red ball. If there were five color types represented in the second bin, then the unknown probability would be 0.2. As more colored balls are added to the bin, the belief that a red ball will be selected decreases. The probability of selecting a red ball will have different weights depending upon the relative atomicity of those red balls.

Atomicity is an element in the equation to determine probability expectation. Because most people are familiar to some degree in dealing with probabilities, probability expectations is a measure of subjective opinions. An opinion is assigned the symbol of "w" in which $w = (b, d, u, a)$, where the components correspond to belief, disbelief, uncertainty, and atomicity. The probability expectation of an opinion (w), denoted by E(w) is defined as follows: $E(w) = b + au$. The probability expectation of a given state consists of the belief, uncertainty, and the atomicity of the state. Atomicity is an element because uncertainty about 'x' atomic states (lowest elements) is split between those 'x' states. (Josang, A., 1999)

In Josang's trust model, he uses an algebra based on the framework for artificial reasoning called subjective logic. Subjective logic uses posteriori probability functions, second order Bayesian probability density functions, and the Bayesian consensus rule to define the various logical operators for combining opinions. These operators are conjunction, disjunction, negation, recommendation, and consensus. These operators are the same as those found in classical and subjective logics, but they are applied to trust.

Josang has determined a method for measuring trust in a statement, but how does one combine opinions? This is done through subjective logic operators. A conjunction of two opinions combines an individual's opinions on two distinct binary statements into one opinion that reflects the belief in the truth of both statements. If x and y are two distinct statements, the conjunction of the belief in x, represented by $W_x = (b_x, d_x, u_x, a_x)$ and y represented by $W_y = (b_y, d_y, u_y, a_y)$ represents an individual's opinion about both x and y being true.

If we represent the conjunction of an individual's opinions on statements x and y as $W_{x \wedge y}$, then $W_{x \wedge y} = (b_{x \wedge y}, d_{x \wedge y}, u_{x \wedge y}, a_{x \wedge y})$. To determine the conjunction, the individual values of belief, disbelief, uncertainty, and atomicity must be combined for the opinions on both statements. The formulas for computing those values are as follows:

- $b_{x \wedge y} = b_x b_y$
- $d_{x \wedge y} = d_x + d_y - d_x d_y$
- $u_{x \wedge y} = b_x u_y + u_x b_y + u_x u_y$
- $a_{x \wedge y} = (b_x u_y a_y + u_x b_y a_x + u_x a_x u_y a_y) / (b_x u_y + u_x b_y + u_x u_y)$.

A disjunction operation would represent an individual's opinion about statements x or y or both being true. The disjunction operation is represented by the symbol "$\vee$", so $W_{x \vee y} = (b_{x \vee y}, d_{x \vee y}, u_{x \vee y}, a_{x \vee y})$. The formulas to determine disjunction are as follows:

- $b_{x \vee y} = b_x + b_y - b_x b_y$
- $d_{x \vee y} = d_x d_y$
- $u_{x \vee y} = d_x u_y + u_x d_y + u_x u_y$
- $a_{x \vee y} = (u_x a_x + u_y a_y - b_x u_y a_y - u_x b_y a_x - u_x a_x u_y a_y) / (u_x + u_y - b_x u_y - u_x b_y - u_x u_y)$

A negation of an opinion represents the belief that a statement is false. This corresponds to the "not" operator in standard logic. If W_x represents an opinion, $W_{\neg x}$ represents the negation of W_x such that $W_{\neg x} = (b_{\neg x}, d_{\neg x}, u_{\neg x}, a_{\neg x})$. The formulas to determine negation are as follows:

- $b_{\neg x} = d_x$
- $d_{\neg x} = b_x$

- $u_{\neg x} = u_x$
- $a_{\neg x} = 1 - a_x$

Subjective logic can also be used to convey values for recommendation. Recall that trust in humans is the belief that the human will cooperate and not defect. Agent A has an opinion about agent B's willingness to cooperate and not defect. Agent B has an opinion about a statement or proposition x. A recommendation consists of combining agent B's opinion about statement x with agent A's opinion about agent B's cooperation, so agent A can form an opinion about statement x.

Agent A's opinion about agent B's recommendations is represented as $W_{AB} = (b_{AB}, d_{AB}, u_{AB}, a_{AB})$. Agent B's opinion about proposition x is represented as $W_{Bx} = (b_{Bx}, d_{Bx}, u_{Bx}, a_{Bx})$. To express agent A's opinion about proposition x as a result of a recommendation from agent B, the following representation is used: $W_{ABx} = (b_{ABx}, d_{ABx}, u_{ABx}, a_{ABx})$. The symbol '$\otimes$' is used to designate a recommendation. In this instance, the recommendation operation would be represented as $W_{ABx} = W_{AB} \otimes W_{Bx}$. The formulas for the individual components are as follows:

- $b_{Abx} = b_{AB}b_{Bx}$
- $d_{Abx} = b_{AB}d_{Bx}$
- $u_{Abx} = d_{AB} + u_{AB} + b_{AB}u_{Bx}$
- $a_{Abx} = a_{Bx}$

The recommendation operator makes the assumption that the agents do not defect or change their recommendations depending upon whom they interact with. It also makes the assumption that the opinions that are recommended are independent. If a chain of recommenders is needed to gain information about a proposition x, it is assumed that only first hand knowledge is transmitted. If second hand knowledge is passed as a recommendation, opinion independence is violated. Additionally, the order in which the opinions are combined is significant.

Subjective logic uses consensus operators as well. A consensus operator allows two independent agents to form a consensus opinion based upon each agent's individual opinions has concerning a proposition 'x'. An example of a consensus operation is two

referees looking at the same play on the field. Both referees have their own opinions as to what they saw. By combining their opinions, they can produce a better picture of what occurred. A consensus operator serves to reduce uncertainty.

If W_{Ax} represents agent A's opinions about proposition 'x' and W_{Bx} represents agent B's opinions about the same proposition x, then $W_{Ax, Bx}$ represents the consensus opinion. The consensus operator uses the symbol '⊕', so $W_{Ax,Bx} = W_{Ax} \oplus W_{Bx}$. The consensus from agent A and agent B's opinions on proposition x is represented as $W_{Ax,Bx} = (b_{Ax,Bx}, d_{Ax,Bx}, u_{Ax,Bx}, a_{Ax,Bx})$. The formulas for the individual components are as follows:

- $b_{Ax,Bx} = (b_{Ax}u_{Bx} + b_{Bx}u_{Ax}) / k$
- $d_{Ax,Bx} = (d_{Ax}u_{Bx} + d_{Bx}u_{Ax}) / k$
- $u_{Ax,Bx} = (u_{Ax}u_{Bx}) / k$
- $a_{Ax,Bx} = (a_{Bx}u_{Ax} + a_{Ax}u_{Bx} - (a_{Ax} + a_{Bx})\, u_{Ax}u_{Bx}) / (u_{Ax} + u_{Bx} - 2\, u_{Ax}u_{Bx})$

The symbol 'k' represents $(u_{Ax} + u_{Bx} - u_{Ax}u_{Bx})$. Additionally $u_{Ax} = u_{Bx}$, but the values equal 0 or 1.

The consensus operators are both commutative and associative. This means that the order in which the opinions are combined have no significance. This operator also assumes first hand knowledge and opinion independence.

Josang provides an example of how subjective logic can be used to measure the trust in a certificate. In the modern PKI system, a certificate authority issues certificates containing an individual's public key. If agent A knows certificate authority B's public key k_b and certificate authority B knows agent C's public key k_c, then certificate authority B can send agent C's public key to agent A signed by certificate authority B's private key k_{-1b}. Agent A will verify the certificate with B's public key, and if correct, will know that it has received a correct copy of agent C's public key.

Unfortunately, this exchange does not convey A's trust that it has received a correct copy of C's public key. To trust in a certificate, A must have an opinion about the validity of B's public key. Agent A's opinion in the key authenticity (KA) of B's public key is represented as $W_{AKA(kb)}$. Agent A must also form an opinion on an agent's

recommendation trustworthiness (RT), which measures A's trust in B to properly certify other keys. Agent A's opinion about B's ability to certify keys, is represented as $W_{ART(B)}$. Agent A must also evaluate the recommendation of B as to the validity of agent C's public key. Certificate authority B's opinion is represented as $W_{BKA(kc)}$.

To validate the authenticity of the certificate, A must first evaluate the recommendation from certificate authority B. Agent A will combine its opinion of B's key authentication with its opinion about B's recommendation trustworthiness. This will determine agent A's opinion about B's capability as a recommender. The recommendation will be represented as the conjunction of both opinions $W_{AB} = W_{ART(B)} \wedge W_{AKA(kb)}$.

Then Agent A must combine its opinion about B's recommendation ability with B's recommendation about C's public key. Agent A's opinion about agent C's public key will be represented as $W_{AKAB(kc)} = W_{AB} \otimes W_{BKA(kc)}$. (Josang, A., 1998)

Josang has demonstrated the versatility of his model by demonstrating that it is capable of chaining trust and certificate relationships using multiple recommendation operators. His model is also capable of measuring trust along multiple trust paths and combining them into a single representation. The model can be used in decision making situations by assigning utility values to the levels of trust computed by the model.

1. Weaknesses in the Josang Trust Model

Josang's trust model violates Reiter and Stubblebine's second principle that states that the meanings of the model's parameters should be unambiguous. In Josang's model, the values for belief, disbelief, and uncertainty must equal 1 (b + d + u = 1). However, he does not provide any confidence parameters to assist the user in assigning values. Hence, users are given a lot of latitude to interpret what the parameters mean, and how to apply them. Do the trust values of (.4 + .2 + .4) give the user enough information to make a decision? The model forces each user to make a subjective trust decision based only on the trust values. If the results are compared against other results, or compared against an index, the decision making process would be easier, as the values would be less ambiguous.

Although Josang does not like to use discrete values because they only provide a small set of possible trust values, discrete values in conjunction with his model would help resolve some of the ambiguity involved in assigning values. (Josang, A., 1998) For example, strong belief may have a value between .95 and .80. Without a measurement or criteria to use in the assignment of values, the ambiguities introduced by each person's subjective interpretations of the metric will affect the input values and how the output values are utilized.

H. WEAPONS RELEASE AUTHORITY

In the United States Navy, the Tactical Action Officer (TAO) is given weapons release authority by the ship's Commanding Officer. The TAO can fire on the enemy without the Commanding Officer's permission. There are a number of reasons that the TAO is given this authority. The TAO is a watch station that is manned 24 hours a day. If a situation occurs that needs immediate response, the Commanding Officer may not be available to consult. The Commanding Officer may be involved with maneuvering the ship to avoid danger, or he may be taking tactical control of other units. Usually, the Commanding Officer and the TAO will be working together to fight the engagement, but the TAO actually fires the weapons.

If the TAO makes a mistake and either fires a weapon when he should not, or did not fire a weapon when he should, the Commanding Officer and the TAO can be fired (i.e., their careers are ruined). The Commanding Officer trusts the TAO with his career, and the safety of the ship and crew.

There is no formal model for how a Commanding Officer decides whether to trust an individual with weapons release authority. However, many of the variables that Essin used in his model also apply to how weapons release authority is granted.

The first variable to consider is reputation. Reputation can be broken down further into an analysis of traits such as professionalism, attentiveness, demonstration of good judgment, and leadership. In the Navy, reputation is established primarily through the individual's ability to properly stand a watch. If the individual is prepared for watch, anticipates the next course of action, and keeps the watch standing team focused on their

job, the individual will have a good reputation. If the individual shows up late for watch, reacts to situations, instead of thinking ahead, and only demands the minimum from his watch team, his reputation will not be good. The Commanding Officer will be able to determine these traits by observing the individual during numerous watches. He can also obtain trust recommendations from other TAOs and the Executive Officer, who also have first hand knowledge of the individual's watch standing ability.

Another variable is knowledge. To become a TAO, the individual must qualify for the position by obtaining signatures from qualified TAOs. A signature indicates that the individual has demonstrated adequate knowledge to accomplish specific tasks. The qualification book is standardized through out the Navy, and contains all of the tasks that a TAO must master. Once all signatures have been obtained, the Commanding Officer personally tests the individual to ensure that the individual has enough knowledge to become a TAO. The tests can be written, oral, practical, or a combination.

Unlike Essin's trust model, the Commanding Officer will want to see how an individual performs under stress. Different people react differently to stressful conditions. Some work well under stress and others react poorly. War is an incredibly stressful situation. Most people will never have to make decisions in life or death situations. A Commanding Officer can evaluate individuals by observing their performance in stressful work conditions (e.g., just before a major inspection), through drills and training exercises. The Commanding Officer can also observe his performance when the individual stands the TAO watch "under instruction" with another TAO lightly supervising.

1. Weaknesses in the Weapon Release Model

This model is similar to Essin's model except that Essin's model was theoretical and not practical, whereas this model is practical, but is not theoretical. In this model, the majority of the trust decision is made outside of the model by the Commanding Officer. Trust is a function of reputation, knowledge and reactions to stress, but there are also subjective factors that are applied differently by each Commanding Officer. Factors such as experience, judgment, environmental concerns (e.g., war, mounting tension,

peacetime, etc.), and cultural bias used in the decision making process are difficult to quantify or qualify.

The Commanding Officer receives trust recommendations from his staff concerning an individual's qualification for TAO. The Commanding Officer must evaluate the trust recommendations, as well as form opinions about each recommender's recommendation trust (i.e., what level of trust does the Commanding Officer have in the individual to make a recommendation). The difference in this model from the others is that the Commanding Officer can combine his first-hand knowledge of the individual with the recommendations about that individual. In the other models, the user did not have first-hand knowledge of the target.

Another difference in this trust model from the others is that the Commanding Officer is not trying to communicate his trust to another entity. He or she does not have to explain any rationale. He or she is free to use any scale or metric to arrive at a conclusion. Since the Commanding Officer also has first-hand knowledge of the individual, he or she is free to discard any of the recommendations. If he or she desires, no recommendations have to be used in the decision making process. This eliminates the need for recommendation trust.

I. SELECTED MODEL

Each model evaluated differed in its definition of trust, the variables of trust used, and the algorithm used to compute trust. All of the models evaluated had strengths and weaknesses. Additionally, some of the models were better suited to specific areas than others.

When selecting an acceptable and practical trust model a number of factors must be considered. The following are steps we used to evaluate the various models to determine if the trust model is capable of being practically implemented:

1) Review the definition of trust used in the model. In order to understand the trust model, one must understand the model's definition of trust to gain a proper frame of reference. It is important to know whether this is an abstract or explicit definition. If it is abstract, is it broad enough? If it is an explicit definition, how is it applied?

2) Review the input to the model. Define the variables that are used as input to the model. Are these the correct variable to use in relation to the model's definition of trust? Are these hypothetical variables, or can they be used in real world applications?
3) Analyze the output. What is the output? Is the output significant in relation to real world events, or are they theoretical. Can the outputs be measured against some standard or index?
4) Determine metrics used. It is also important to evaluate the metrics to determine the amount of subjectivity in assigning values to variables. Is the model based upon mathematical principles? What algorithms are used in determining an output? How interoperable is the model? Are metrics applied uniformly? Does the metric take into account as much information as is necessary to make a decision?
5) Evaluate the environment to which this model is applied. This is especially important if an explicit definition of trust is used. Again this is important to gain a frame of reference in which to evaluate the model. This is also important in measuring a model's applicability to real world scenarios.
6) Identify assumptions made in the model. Are the assumptions valid? Can the model be applied to real world situations given the assumptions? How do the assumptions affect the model?
7) Identify strengths and weaknesses of each model. Does the model accomplish its objective? How robust is the model? Can it be applied to practical situations, or is it a theoretical model designed to stimulate thought in an area? Are there items or concepts missing from the model that should have been addressed? Is there functional interoperability? Can the model transfer trust between entities?

The first and most important consideration is whether the model can be implemented in the real world. Models such as Essin's trust model, which are more theoretical than practical would not be considered for implementation. Additionally, if too many paradigms or assumptions are required to implement the model, as in Reiter and Stubblebine's model, then those models will not be considered for practical use.

Practical trust models must be also be scalable to the Internet. PGP has already been implemented, but it is not scalable enough for e-commerce or transactions involving a large number of entities.

The TAO model is another practical implementation of a trust model, but it is not the correct model to apply to the DoD PKI system. The TAO model is not a formal model that uses static input variables, an algorithm and a measurable output. The user can add additional trust variables, as he desires. Additionally, the model cannot be automated because most of the trust decision is made outside of the model.

The Abdul-Rahman and Hailes trust model was very good, but a flawed algorithm precluded a meaningful output. As a result, it is not suitable for use with the DoD PKI system.

As a result of the analysis of the models reviewed, we felt that the Audun Josang trust model was the most comprehensive of the models evaluated and it had the greatest potential to be practically implemented. The model uses a powerful algorithm that combines subjective values with establish mathematical principals. The model was flexible, the variables used correlate to the DoD PKI system, and the model can be automated with little input from the user. Therefore, the Josang model will be used to evaluate the implementation of a trust model to the DoD PKI system.

V. PUBLIC KEY INFRASTRUCTURE (PKI)

A. PUBLIC KEY CRYPTOGRAPHY

A public key infrastructure (PKI) is the key management system that ensures that authenticated public keys are safely, efficiently, and conveniently delivered to the system that needs them. PKI tries to provide the interoperability, standardization, policy, and trust necessary to manage key pairs and certificates. To understand how the PKI system works, one must first understand public key cryptography.

We have already discussed the fact that cryptography can be used to attain various levels of confidentiality, authentication, integrity, and non-repudiation, in combination with other techniques. Encryption combines plaintext data and an encryption key. Decryption requires the use of a decryption key to be able to retrieve the original plaintext data. A key is a random string of bits, whose length depends on the type of cryptographic algorithm being used and the desired level of strength against attacks on the key. There are two types of cryptography: symmetric and non-symmetric.

In symmetric cryptography, the same key that is used to encrypt the message is used to decrypt the message. This means that the user and the recipient must be in possession of the same key. User A encrypts a message with key (K_S) and sends it to User B. User B must have key (K_S) to decrypt the message. If only user A and user B have the key, then the information is secure. Common symmetric cryptographic algorithms include the following: DES, SKIPJACK and CAST-128.

The primary weakness of symmetric cryptography is the difficulty in distributing the keys. Since both users must use the same key, a secure method of distributing that key must be devised. Additionally, both users have to form a number of opinions concerning the validity of the key being used. The users must form an opinion on the security of the method used to distribute the keys. Both users must also have an opinion on the trust they place in the validity of the keys that they are using. Since both users

share the same key, they must also form opinions on the trust that they have in the other user to properly safeguard the key.

Scalability is also a problem. In actual practice, the same key is distributed to many users for interoperability reasons. Unfortunately, this is a brittle system, because a single compromised key can destroy the entire system, at least until the life span of the key.

Another weakness of symmetric cryptography is that it cannot be used for digital signatures without the use of a third party arbitrator. One of the requirements for a digital signature is that the key used is unique to the owner. In symmetric cryptography, more than one person holds the same key. It is not possible to determine exactly who signed the message by examining the key. An arbitrator can be used for digital signatures, but this method involves considerable overhead and trust in the arbitrator. In this method the symmetric key held by user A and an arbitrator is used to sign a message. The arbitrator confirms that only A has the corresponding symmetric key.

Asymmetric cryptography, or public key cryptography, was developed by Whitfield Diffie and Martin Hellman to resolve the key distribution problems associated with symmetric cryptography. The Diffie-Hellman algorithm is used primarily for key distribution. Another more common public key algorithm is Riverst-Shamir-Adleman (RSA). RSA can be used for key distribution, encryption and decryption, and digital signatures.

Asymmetric cryptography uses a pair of related keys to perform cryptography. When the keys are generated, one is designated the "private key", which is kept secret and the other key is the "public key", which is available to everyone. To be commercially acceptable, it must be computationally infeasible to determine the decryption key given the cryptographic algorithm and the encryption key.

Either key can be used to encrypt a plaintext message and its related key can be used to decrypt it. Tom can encrypt a message using his private key, which only Tom possesses, and sent it to Linda. Linda can decrypt the message with Tom's public key.

Conversely, Linda can send an encrypted message to Tom, using Tom's public key. Tom would then use his private key to decrypt the message.

When Tom uses his private key to encrypt a message, he is ensuring authentication and integrity of the message only. Encrypting the message with a private key will not provide confidentiality, because anyone can obtain the public key and read the message. However, if Linda decrypts a message with Tom's public key, she knows that only Tom could have encrypted the message: Tom is the only person with the private key. Thus Linda can confirm the integrity and authentication of the message.

Public key cryptography allows a user to digitally sign a plaintext message. Because only Tom has the private key, any message encrypted with Tom's private key must have been generated by Tom. Therefore, the encrypted message acts as a digital signature. Although encrypted messages can act as digital signatures, another method is typically used in order to save storage space. Generally a hash (also known as a digest) or message authentication code (MAC) is used to make a fingerprint or checksum of the message. If the message is altered in any way, the hash or MAC will not equal the altered message. Once a hash or MAC is generated, Tom will encrypt it with his private key. Linda will use Tom's public key to determine the hash or MAC and will generate a hash or MAC of the message to compare the two values. If the values are equal, then the message's integrity has been proven. If they do not match, the message has been altered. Linda can authenticate that Tom sent the message, because only Tom has the private key. This digital signature also provides non-repudiation, in which Tom cannot deny sending the message. If the message is digitally signed with Tom's private key, then only Tom could have sent the message.

B. KEY DISTRIBUTION

Key management with symmetric cryptography can be very difficult because both keys must be kept secret. Since both the sender and recipient must have the same key to encrypt and decrypt information, a method must be devised to securely distribute the keys. A standard ANSI X9.17 was developed to address symmetric key distribution among financial institutions. This standard identified three types of symmetric keys. The

first type of key is known as a "session key," which is used to encrypt the majority of message traffic. The next key is known as a "key-encrypting key." When new session keys are needed, a message is sent to the intended recipient, containing the new session key. To protect the session key in transit, key-encrypting keys are used to encrypt the session keys. The final type of key is called a "master key." This master key is used to encrypt session keys or key-encryption keys in transit. The master key is manually distributed.

There are two types of X9.17 configurations. The first is a point-to-point configuration in which two users share a master key. User A generates a session key or a key-encrypting key and sends it to user B, encrypting the key with the master key that both users share.

Unfortunately, if a number of people want to communicate with each other using this system, the number of keys necessary can become overwhelming. If N users want to use this system, N^2 master keys are needed. Additionally, if a master key is ever compromised, all session keys and key-encryption keys are also compromised.

Another X9.17 configuration involves a key management center. In this configuration, each user shares a master key with the key distribution center, but not with each other. When user A wants to communicate with user B, it requests a session key from the key distribution center. The key distribution center generates a session key and returns it to user A, encrypted with the master key that both share. It also sends a duplicate session key to user A, however this session key is encrypted with the master key that is shared between the key distribution center and user B. User A retains the session key encrypted with A's master key and forwards the other session key to user B. This method only requires N master keys for N users. However, this configuration requires complete trust in the key distribution center. (Ford, W., and Baum, M., 1997)

Another way of distributing symmetric keys is through the use of public key cryptography. User Z encrypts a message containing an identifier of Z and a nonce (N_1) with user B's public key. B sends a message to Z containing a new nonce (N_2) and the original nonce (N_1). The original nonce verifies that the recipient of the message is user

B and not a malicious entity, because only B could decrypt the message. Z sends back (N_2), so B can authenticate Z. Then Z generates a secret symmetric key and sends it to B, encrypting it with B's public key.

Another variation is a hybrid approach. The message is encrypted using a symmetric session key and the session key, encrypted with the recipient's public key, is appended to the message. When the recipient receives the message, the session key is decrypted using the recipient's private key and the message is then decrypted using the session key. (Stallings, W., 1999) Secure/Multipurpose Internet Mail Extension (S/MIME) uses this form of key distribution.

Exchanging keys in public key cryptography is different from symmetric key distribution, because only the private key must be kept secret. The public key can be freely disseminated. If a key is compromised, user A only has to generate a new key pair and redistribute the public key. The difficulty in distribution of public keys is authenticating the public key. How does one determine if a public key really belongs to Alice, and not a malicious agent? Oscar, a malicious entity posts a public key on a bulletin board under the name of Alice. Oscar then sends a message to Dave pretending to be Alice. Oscar also digitally signs the message with the private key associated with the public key on the bulletin board. When Dave verifies the digital signature using the public key on the bulletin board, he will think that the message was from Alice.

There are a number of ways to distribute public keys, but they all fall into four general categories. The first method is for the user to simply send his or her public key to anyone the user wishes to communicate with. This is similar to the way keys are exchanged in PGP. The weakness with this method is that a malicious entity can easily distribute messages containing a public key pretending to be user A. This method does not authenticate the sender of the message, nor does it scale well.

Another method is to use a public directory. The directory authority maintains the directory, which includes the name and public key of each subscriber. Each user would register a public key in person or through some authentication method. Anyone wishing to use a subscriber's public key can obtain it from the directory. The public key will be

signed by the directory authority to verify that the key came from the directory authority. This method is more secure than the previous method, but it still has vulnerabilities. If a malicious entity can compromise the directory authority's private key, all of the subscriber's keys have been compromised.

Another distribution method is called public key authority. It is similar to using a directory authority, but the authentication among the authority and the users is greater. This method uses time stamps and nonces to ensure that there are no man-in-the-middle attacks by malicious entities. This system is more secure, but still has the same vulnerability as the previous method.

The final method of key distribution involves the use of certificates. A certificate contains the name, attributes, and public key of a user. A certificate authority (CA) generates the certificates. When a certificate is issued, the CA verifies the identity and attributes of the user. A certificate is generated and is digitally signed by the CA to vouch for its authenticity and integrity. To distribute a public key, the user sends his certificate to the intended communication partner, or the partner can obtain the certificate from the CA. The recipient of the certificate verifies the CA's digital signature, and is then free to use the public key in the certificate. The primary benefit of certificates is that a user only needs the CA's public key to obtain the public keys of anyone in the CA's domain.

C. CERTIFICATES

There are three distinct types of certificates that are grouped according to their use. The first and most common is an identity certificate. This certificate is designed to bind an identity to a public key. The subject of an identity certificate is any entity that can be designated with a name. It can be a person, personal computer, server, or sensor. The subject named in the certificate is the entity that is using the private key that corresponds to the public key in the certificate.

The second type of certificate is called an attribute certificate. This type of certificate is not concerned with the identity of an entity. An attribute certificate lists the characteristics of an entity that detail what actions that entity is permitted to take. Within

the PKI community, some feel that it is more important for a certificate to list a user's privilege levels for specified applications, than authenticating an identity. The security policies embedded in some applications are not concerned with who the user of the system is, so long as the user has the appropriate attributes (security level) for the application that is being executed. Proponents for attribute certificates also point out that many naming schemes are limiting. It is not uncommon for two entities to have the same name (name collision). This problem can be exacerbated in some societies that have large populations with the same surname, like in China, Korea, and Wales.

The third type of certificate is a cross-certificate. When all communication is transacted in a certificate authority's (CA) domain, then all users know the mechanisms that the CA uses to authenticate an entity. When transacting business within a domain, a certain level of trust can be placed in the certificate, because of the reputation of the CA. If the CA required three types of identification and screened the individual's social security number through the social security administration, then trust in the certificates will be high. If the authentication was done using an e-mail address, then there is little trust in the certificate. When dealing with certificates outside of the domain, the user must obtain the public key of the other domain's CA to verify the certificate. However, the user does not know anything about the business practices of the out-of-domain CA. A cross-certificate is issued by the domain CA and contains the identification and public key of the out-of-domain CA. A cross-certification is only issued if the other CA meets the standards of the domain CA. In this way, cross-certificates transfer a degree of trust from one domain to another.

Cross-certificates also validate the authenticity of the public key of the out-of-domain CA. If user A receives a certificate from another CA, and user A does not have a secure copy of the other CA's public key, then that certificate is worthless. A cross-certificate provides a secure copy of the out-of-domain CA's public key; because the digital signature on the certificate can be authenticated with A's secure copy of the public key of the issuing CA.

Political, business and sociological concerns will ensure that there will never be a single CA. As a result, a user often receives certificates that were not generated by his CA. In this case a user must obtain a secure copy of the public key of the CA issuing the certificate. The user must determine if its CA has a cross-certification with the other CA. If a cross-certificate exists, then the user can easily obtain the public key. However, when a cross-certificate does not exist, the user must find a certificate path to obtain the public key of the certificate issuing CA. This may necessitate obtaining cross-certificates from CA(a) to CA(b), then obtaining the cross-certificates from CA(b) to CA(c). In this way, a certificate path is established from CA(a) to CA(c).

D. X.509 STANDARD

The certificate standard that is most common is the X.509 standard. There are a couple of other certificate standards (e.g. PGP certificates and Simple Public Key Infrastructure (SPKI) certificates), but they have not been commercially accepted.

In 1988, the International Telecommunication Union (ITU-T) developed the X.509 certificate standard as part of its X.500 series of directory standards. The X.500 standard defined a hierarchical directory system that was to list the X.509 certificates. An access language called the Lightweight Directory Access Protocol (LDAP) was used to query the X.500 directory to obtain the certificates.

In 1993 X.509 version 2 (X.509 V2) was released. It added additional fields to help directory access. However, when commercial applications using X509 v2 were being developed they found that the structure was too restrictive. One of the largest problems was that the X.500 name, which was used to name the entity owning the certificate, needed to be expanded to prevent name collisions.

As certificate technology expanded, more fields became involved in developing the standard. In 1996, standards organizations (ITU, ISO/IEC and ANSI X9) developed X.509 v3. This version added optional extension fields to the format. The X.509 v3 format is used in a number of commercially accepted products including: S/MIME, IP security and Secure Socket Layer (SSL). The format of the X.509 v3 standard is provided in figure 1.

Version (of Certificate Format)		
Certificate Serial Number		
Signature Algorithm Identifier (For CA's Signature)		
Issuer (Certification Authority) X.500 Name		
Validity Period (Start and Expiry Dates/Times)		
Subject Name (X.500 Name)		
Subject Public Key Information	Algorithm Identifier Public Key Value	
Issuer Unique Identifier		
Subject Unique Identifier		
Extension Type	Critical/Non-critical	Extension Field Value
Certification Authority's Digital Signature		

Figure 1. X.509 v3 Certificate

The certificate has the following elements:

- **Version**: This lists the version of the certificate that is being used. In most cases this will be version 3.

- **Serial number**: An integer value assigned by the CA that uniquely identifies the certificate.
- **Signature algorithm identifier**: This field identifies the algorithm used to digitally sign the certificate and any related parameters. In reality, this field is seldom used because the same information is contained in the field associated with the CA's digital signature.
- **Issuer name**: X.500 name of the CA that created and signed the certificate.
- **Validity period**: This field lists the start and expiry dates of the certificate.
- **Subject name**: This is the name of the entity that corresponds to the public key. Previous versions only allowed X.500 names, but version 3 allows additional naming formats.
- **Subject public key information**: This field lists the algorithm, parameters and public key of the entity in the subject name field.
- **Issuer unique identifier**: This is an optional field. This field contains additional information about the CA. This may be needed if the X.500 name is not concise.
- **Subject unique identifier**: This field is also optional. This field is also used to provide additional information about the subject to ensure name collisions do not occur.
- **Extensions**: This field is also optional. The X.509 v3 standard supports numerous extension fields depending on the application's need. An application may need additional extensions to support additional security. The PKIX working group has standardized common extension types. However, applications are free to develop their own extensions.
- **Issuer's signature**: This field contains the algorithm identifier, parameters and the certificate hash signed by the CA's private key.

In addition to the standards organizations, a working group called the PKIX Working Group of the Internet Engineering Task Force (IETF) was formed to address standardization, security, and interoperability issues with X.509 v3. It developed the RFC 2459 profile that further specified how the standard was to be implemented.

E. CERTIFICATE REVOCATION LIST (CRL)

When an individual receives a certificate, he verifies the certificate by decrypting the certificate's hash code, thus verifying the integrity of the certificate and the authenticity of the CA. However, this would still not alert the user of the certificate to a situation where an individual's private key has been compromised. If a company issues a certificate to an employee, and the employee leaves the company, how does the company alert its clients not to use the certificate anymore?

The X.509 standard also set up a format for a certificate revocation list (CRL) that allows a user to check if the certificate is still valid. The CRL lists all of the certificates under a CA that were revoked. The format of the X.509 CRL v2 is shown in figure 2.

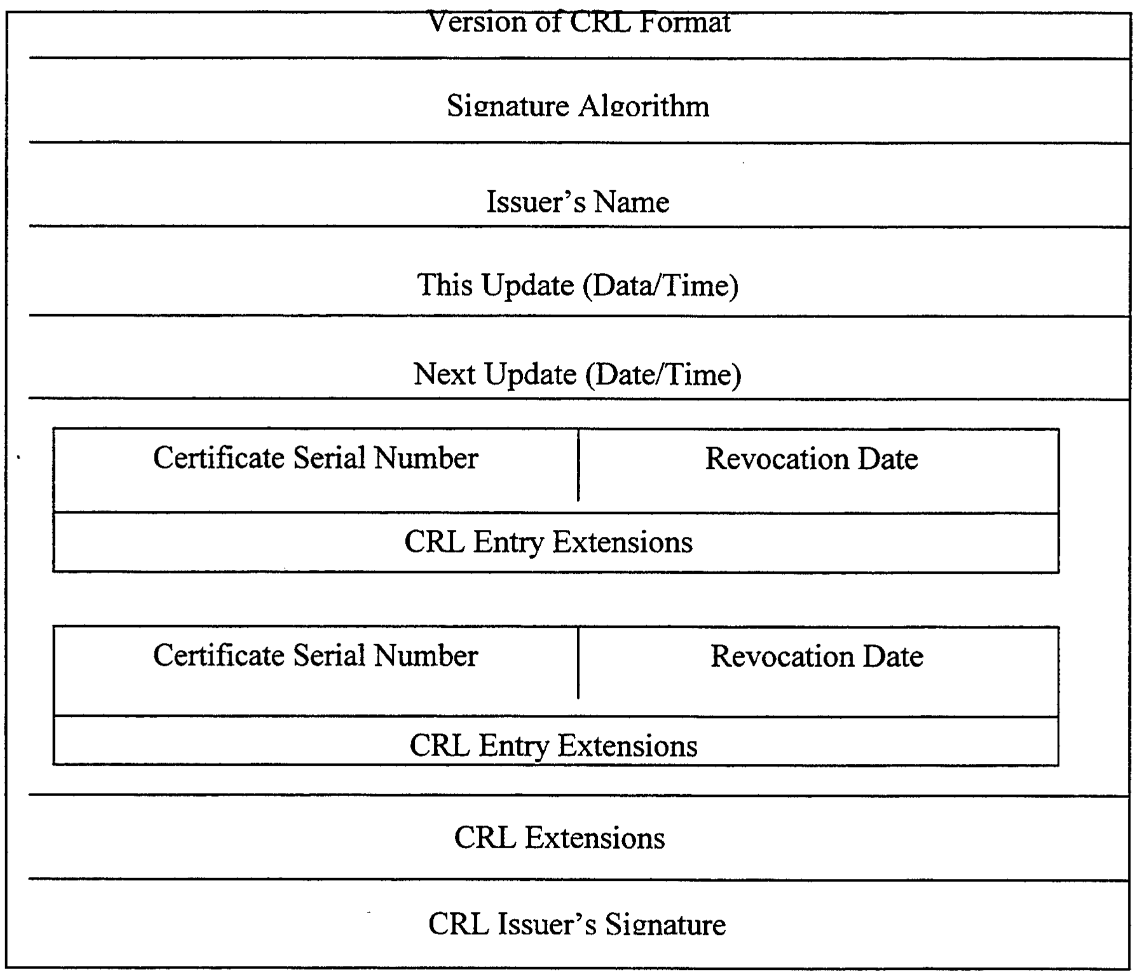

Figure 2. CRL v2

The elements are as follows:

- **Version**: Indicates the version of the CRL format. It will be either version 1 or 2.
- **Signature algorithm**: List the algorithm used to sign the CRL
- **Issuer's name**: Name of the CA that issued the CRL. This is an X.500 formatted name.
- **This update**: Date and time that the CRL was issued.
- **Next update**: This is an optional field. It is used to list the date and time that the next CRL will be issued.
- **User certificate**: This is the serial number of the certificate that has been revoked.
- **Revocation date**: Lists the date that the certificate was revoked.
- **CRL entry extensions**: This is an optional field. It holds additional information. Each type is also standardized by the PKIX group.
- **CRL extensions**: This is also an optional field. This contains information that pertains to the entire CRL.
- **CRL issuer's signature**: This is the digital signature of the CA issuing the CRL.

There are a number of reasons a certificate can be revoked. A person might need to change his key pair (excessive use of a key gives a hacker more material to break the key), an attribute on the certificate may need to be updated, or a private key was compromised.

Anyone using a certificate should check the CA's revocation list in addition to validating the digital signature before using a certificate. CRLs are generally listed in a directory system managed by the CA. Most of these directories use a X.500 structure. The CRLs also list the reason a certificate has been revoked.

One of the problems with CRLs is their size. As certificates are revoked, the entries to the CRL increase. Revoked certificates have to remain on the CRL until their expiration date. The size of a CRL can easily become unmanageable, and difficult to use. Fortunately, CRL v2 contains provisions that allow the CA to partition the CRL into distribution points, which reduces their size. Delta-CRLs were also proposed. Delta-

CRLs only list the revoked certificates since the last update. Indirect CRLs are another method to improve CRLs. In an indirect CRL, an entity other than the CA lists the CRLs from multiple CAs. This reduces the number of CRLs that must be fetched to verify a certificate path.

Another problem is the speed at which the CRL is updated. CRLs are issued on a regular basis. They are not real time. As a result, there is latency in the CRL. Online Certificate Status Protocol (OCSP) was developed as an alternative to address the latency problem, but it adds additional technical and security risks.

F. CERTIFICATION AUTHORITY (CA)

The certification authority plays an integral role in the PKI process. The CA performs the administrative functions of the PKI. The most important attribute of a CA is trustworthiness. The CA performs numerous functions that affect people's perception of their trustworthiness. If a CA is not trusted, then none of the certificates that the CA issues are trusted.

The first function that a CA performs is validation. The CA is responsible for accurately identifying an individual or entity. The level of certificate issued determines the extent that an individual's identity is verified. CAs normally determine identity by verifying passports, drivers licenses, credit card statements, and social security numbers. If high security is required, fingerprints, signature cards and other authentication methods may be used. In many cases the actual identity verification is delegated to another organization called registration authority (RA). The validation is essential to establish trust in the name-to-certificate binding.

The CA is also responsible for creating and posting certificates. The CA is responsible for ensuring that public and private key pairs are generated securely with the strongest possible algorithms and key length. Once the key is generated, the private key is securely transported to the entity requesting the certificate. Another option is to have the key pair generated in the user's system. The public key is then securely transferred to the CA, and the private key remains on the system. This is preferable for certificates used for digital signatures. (Ford, W. and Baum, M., 1997)

Once an individual has been authenticated and a key pair has been generated, the CA builds a certificate. The certificate will contain an individual's attributes and the corresponding public key. It will also contain any additional information necessary for its intended use. The certificate will also have an expiration date that is assigned in accordance with the CAs policy. The CA then digitally signs the certificate, verifying the authenticity and integrity of the certificate. The certificate is then posted in a directory managed and updated by the CA, which is accessible to the public.

The CA's private key must be safeguarded from malicious entities. If the private key of the CA is compromised, all of the certificates it generated are no longer trustworthy. As a result, the CAs must use every precaution to safeguard its private key.

The CA is also responsible for key-pair updates. Cryptographic keys cannot be used indefinitely. Good security policies dictate that the keys be changed periodically. This is due to the fact that the more encoded text that a hacker can obtain; the easier it is for them to break the encryption key. By changing the key periodically, it reduces the likelihood of a hacker cracking the key. Additionally, if a key has been compromised without anyone's knowledge, then once a key exchange has been performed, the compromised key is no longer of any use. When a key pair needs to be updated, a new key pair and certificate are generated. The frequency of the key updates is therefore of paramount importance to the security of the system.

In some cases the CA acts as a key escrow agent. If a private key is used only to decrypt files and correspondence encrypted with the public key, then the private key must be held in escrow. If the person encrypting files is an employee and that employee leaves with the private key, the encrypted files may never be decrypted. Another reason key escrow is important is the case when a password protecting the private key has been forgotten, or a private key was held on a floppy disk that developed errors and could not be read. A backup copy of the private key is necessary. CAs can provide this service. If the key pair is used for digital signatures, then the private key cannot be held in escrow, as it would not be legally binding. This is the reason that many people have two certificates. One is used for encrypting files and the other is used for digital signatures.

The CA is also responsible for revoking certificates. If a certificate's private key has been compromised, or the user needs to update a key pair before the certificate's expiration date, the CA must revoke the certificate. The CA must add the revoked certificate to the CRL in addition to ensuring that the CRL is accessible to anyone who needs it. As discussed earlier, the CA can use a CRL, delta-CRL or OCSP to advertise revoked certificates.

CAs are primarily responsible for key and certificate generation and management. However, in some small domains, the CA can also act as an archivist, storing certificates past their expiration dates. For legal purposes it may be necessary to store certificates used for digital signatures, just as it is necessary to save signed contractual documents for a period of time. CAs can also be used for key destruction, to ensure that old keys are not exploited. (Hansen, A., 1999)

CAs are the foundation of the PKI system. If people trust that the CA's public key they hold is valid, and they trust that the CA accurately validates the identification of the individuals to whom they assign certificates, then they will have trust in the certificates the CA issues.

G. PKI MODELS

Simply stated, PKI is a system for publishing the public keys that are generated in public key cryptography. The PKI must not only ensure access to public keys, but it must also provide a level of trust that the name-to-key bindings are valid. There are a number of different models (see glossary for definition) that have been proposed to address these issues.

As stated earlier, the easiest PKI solution to implement is to provide for only one CA. This would avoid all of the interoperability and standardization issues. In addition the CA's public key could be embedded in hardware. However, this is not a realistic solution. Social and political forces will likely discourage the formation of a single CA. The model also has problems regarding the registration of individuals, the CA changing its key pair, and the fact that hackers need only concentrate on hacking one key to cripple the PKI system.

Another version of the single CA model is to incorporate RAs. The CA still acts as the trust anchor, but the responsibility for registering individuals is delegated to the RA. This solves part of the problem with registering individuals, but it does not solve any of the problems that are inherent with a single CA model.

A problem that is common to all of the PKI models that use multiple CAs, or use RAs to register and issue certificates, is that transitive trust must be considered. Recall that transitive trust is when Joe trusts Mary, and Mary trusts Sue, then Joe trusts Sue. Unfortunately, trust is not transitive. So, when the PKI models utilize multiple CAs and RAs, the complexity of the trust decisions that the user must make increases. These problems compound when cross certifications outside of a CA's original domain occur. For example, each of our Allies has a hierarchical CA structure, but we do not want to cross certify the same way with each of our Allies, because we have separate bilateral security agreements with each Ally. The desire to limit access to a domain greatly complicates the cross certification process.

The model used with current browsers consists of an oligarchy of CAs and delegated CAs. Instead of one key being hard wired into a system, current browsers have a set of public keys belonging to different CAs called "configured CAs" loaded into the browser software. These CAs can also delegate certificate generation and validation to other CAs called "delegated CAs." Both the configured CA and the delegated CA are trusted and a certificate from either CA will work on the system. To validate the configured CA's public key, the user follows the certificate chain from the delegated CAs to the configured CA. This system makes it easier to obtain certificates, but a compromise in the configured CA's private key still cripples the system. (Perlman, R., 1999)

A hierarchical structure or a top-down architecture[1] is another common PKI model[2]. In this model there is one CA that acts as the root CA, or the trust anchor. The root CA delegates certificate generation and registration to other CAs, who in turn can delegate that responsibility to other CAs. This model is similar to a tree diagram where

[1] Architecture in this context refers to the physical and logical configuration of the CAs, RAs, and users.

each node is represented by a CA. In the military the root CA might be NSA. The NSA CA could delegate to the DOD CA, who could then delegate to the Navy CA. The Navy CA would be responsible for issuing certificates and registering individuals. The certificate they issue would be in the form Jim @Navy.DoD.NSA. In the top-down model, users know the root CA's public key (obtained securely using an out-of-band method, or hard wired in their system). When Jim sends his certificate to Mary, he can also send the public keys for CAs DoD and Navy. Since Mary already has the root CA's public key, she can easily authenticate the certificates from DoD and Navy and verify Jim's certificate. Unfortunately, this type of model also has a single point of failure in that the security depends on a single root CA's private key. Additionally, everyone under the root CA would have to abide by that CA's policies, which may not suit some organizations.

Another PKI model employs a mesh architecture in which there is no root CA. Each CA is free to use its own policies and security practices in issuing certificates. Each CA then cross certifies each other to form certification paths, or a "web of trust." This architecture significantly reduces the effect produced if a CA's private key was compromised. Cross certification makes it easy to follow a certificate path through multiple domains, but dealing with revocation lists may become overly burdensome. This architecture also scales well within the Internet. As new domains and CAs are created, they can join the model, by cross certifying with another CA that is already in the model. However, lack of a central authority requires that the CAs police themselves to ensure that they are properly authenticating the keys of individuals.

The Up-Cross-Down PKI model uses a variation of the mesh architecture. This model combines a hierarchical structure with a mesh architecture. The model assumes that a number of CAs exist, each containing a number of subordinate CAs. In this model a root CA is considered the parent. The parent certifies the keys of the child CAs. The child CAs also certify the key of the parent. The relation between the parent and child is analogous to the 'up' and 'down.' The 'cross' relationship occurs when two root CAs

[2] In this context model is a simplified description of the PKI system, of which architecture is a subset.

cross certify each other. Assume that Joe in the Air Force wants to certify a certificate received from Tom at Lockheed. Joe would first check his parent CA the DOD to see if a cross certificate existed between the DOD CA and the Lockheed CA. If a cross certification did not exist, Joe would continue going up the certification path to the parent NSA. Assuming NSA did have a cross certification agreement with Lockheed, then Joe could validate the certificate path. The major disadvantage to this system is that the user must follow a rigid up-down-cross path. This structure may create a very long certification path, which may cause unacceptable delays in completing requests while validating the path.

A solution to this problem was developed in the Flexible Bottom-Up model. This model is similar to the up-down-cross model, but it allows more paths. This model allows certificates to be issued that can include, add, or exclude branches in the trust chain. It is possible to bypass the traditional up-down-cross path by having individuals in one path cross certify with another CA to shorten the trust chain. This model allows greater flexibility than the up-down-cross model, but the added flexibility has the potential to be abused by everyone wishing to shorten his or her trust path. Abuse would lead to a PKI system like PGP, which works on an anarchy model. (Perlman, R., 1999)

Another mesh model concerns a third party CA that acts as a bridge between the various CAs. Instead of the CAs cross certifying with each other, they cross certify with a third party "bridge CA". The bridge CA only issues certificates to other CAs. This architecture offers numerous advantages. When the bridge certifies a CA, it will gather data on the algorithms used, the CA's architecture and protocols used. The bridge can provide CA configuration data so other CAs can determine interoperability with that CA. In order to provide interoperability, the bridge CA may be able to act as an intermediary between two CAs with differing protocols and algorithms. The bridge CA can also perform functions such as time stamping, which is very important for non-repudiation. Additionally, the bridge CA can have a central repository of CRLs for all of the participating CAs. This would make certificate validations much faster and more convenient. Although a compromise of the bridge CA's private key would cause

disruption, it would not have the same effect as if a root CA's key was compromised. In this case the bridge CA would revoke the old key and could create a new key pair and issue new certificates to the CAs. Each CA would still have security within its domain.

H. DOD PKI SYSTEM

The Assistant Secretary of Defense for Command, Control, Communications and Intelligence (ASD(C3I)) is the Policy Management Authority (PMA) for the DoD's PKI. Under that office is the DoD PKI steering committee, which deals with the technical aspects of the DOD's PKI. The National Security Agency (NSA) provides the DoD PKI Program Manger, who is responsible for developing, procuring, and implementing the PKI architectures and security concerns. The program manager is also responsible for ensuring interoperability among the DoD, government agencies, NATO, and commercial partners. The Defense Information Systems Agency (DISA) provides a deputy program manager and is responsible for integration of centralized components such as CA servers and directory services.

The goal of the DoD PKI is to provide a PKI that can support commercial standards, provide the necessary level of security, while at the same time providing for secure interoperability with other government agencies, Allies, and commercial partners. Another goal is to minimize the effect and overhead the PKI will have on routine operations. The DoD PKI is still in developmental stages. However the DoD is committed to utilizing commercial products and standards to the maximum extent possible to achieve interoperability, yet still provide the level of security necessary. The DoD recognizes that PKI is still an immature technology, so it is committed to actively participating with industry to take advantage of early adoption of technology and services. (PKI Roadmap for the DoD, 1999)

The DoD PKI uses a centralized hierarchical architecture with decentralized RAs. The PKI will also abide by the defense-in-depth, layer information assurance (IA) specifications. (PKI Roadmap for the DoD, 1999) Identifying the value of the information is critical to choosing the level of security that the PKI system must have to protect that information. To achieve this end, the DoD PKI system will have to support

three levels of assurance. Class 3 and class 4 assurance levels correspond to medium and high security. Class 5 assurance level provides the highest level of security, corresponding to classified information transmitted in a high-risk environment.

Class 3 and class 4 PKI systems have been deployed as pilot projects, but are still in developmental stages, and have not yet been deployed extensively. The class 3 system uses a strict hierarchical architecture with the NSA acting as the root CA. The root CA issues certificates to subordinate CAs run by DISA. DISA registers RAs, who in turn register end users. After registration, the user authenticates with the appropriate CA, generates a key pair and securely transfers the public key to the CA. The CA then issues a certificate to the end user. Class 3 systems use the X.509 v3 certificate and the X.509 v2 CRL, which is published weekly (latency is less than 24 hours from revocation). The certificates are registered with the Defense Mega-centers in Chambersburg, Pennsylvania and Denver, Colorado. Class 3 uses RSA, SHA-1 and triple DES as its encryption algorithms. All DoD users are supposed to have class 3 certificates by October 2001. To improve security, the class 3 PKI system is going to start migrating to class 4 by January 2002. (Hansen, A., 1999)

The class 3 PKI system is operational, but it does not support cross certifications. To interoperate with the class 3 system, an outside CA must be certified by the DoD, support the DoD policies, and it cannot act as a trust anchor for the domain. This is called "assimilation." However, many commercial entities and countries may not be willing to abide by this structure. Additionally, interoperability with Allies is a problem because triple DES and certain key sizes are not allowed outside of the United States. The Allies also present political problems as we have separate bilateral agreements with each of our Allies. We may not want to grant some of our Allies, or any of our Allies, the access to our domain that assimilation would grant.

The class 4 PKI system requires that an individual's private key be stored in a cryptographic hardware token. As a result of this requirement, class 4 PKI uses a FORTEZZA Cryptographic card that is integrated into the Defense Messaging System (DMS). The class 4 PKI system was part of the NSA's Multi-Level Information Systems

Security Initiative (MISSI). When it was developed, PKI standards were in their infancy, so the government applied its own standards.

The class 4 PKI architecture is also hierarchical. The root CA, known as the Policy Approval Authority (PAA), is also NSA. The PAA certifies subordinate CAs known as Policy Creation Authorities (PCAs). PCAs certify third tier CAs, who ultimately issue certificates to end users. The CAs use RAs called Organizational Registration Authorities (ORAs) to register end users.

The class 4 PKI will use X.509 v3 certificates and X.509 v2 CRLs. The revocation policy requires a daily update and latency of less than 6 hours after notification. The class 4 PKI repository uses a X.500 directory structure. Additionally, class 4 PKI systems use FORTEZZA, which supports FIPS-approved cryptographic algorithms including SHA-1, DSA, and SKIPJACK.

The class 4 PKI is designed to support access control and privilege management as well as digital signatures. The certificate can be used for validating security levels, verifying message release authority, and file access.

Since the class 4 PKI uses FORTEZZA, anyone wishing to interoperate with the class 4 PKI must convert their system to FORTEZZA. This would require assimilation, which presents the same problems as those in the class 3 PKI systems. Cross certification is possible with FORTEZZA and X.509 v3; however, there are problems with data confidentiality outside of the PAA's domain. (Hansen, A., 1999)

The DoD recognizes that FORTEZZA does not provide the interoperability needed in a PKI system. As a result, the DoD is working on a new class 4 PKI, known as the "target PKI." The new PKI system still uses a hierarchical structure that has the NSA as the root CA. The root CA delegates to regional CAs that utilize RAs to register end users.

The target PKI will use commercially accepted standards including X.509 v3 certificates, X.509 v2 CRLs, and possibly OCSP depending upon its commercial acceptance. All cryptographic algorithms will be FIPS compliant.

The certificate repository has not been defined. The X.500 standard will probably be used in conjunction with LDAP, but X.500 is not commonly used in the commercial sector. If a better directory system is developed, then the DoD might employ it, but nothing has been developed to replace the X.500 standard at this time.

The target PKI will use assimilation instead of cross certification when it initially deploys. Until the PKI standards mature, assimilation guarantees interoperability. However, DoD realizes that to accomplish its goal of interoperability, it must eventually adopt cross certification or use a bridge CA. (PKI Roadmap for the DoD, 1999)

The class 5 PKI system is not available in a prototype at this time. Specifications have been outlined, but current emphasis is on fielding the target PKI system. Since the class 5 system is still in developmental stages, this paper will not discuss its possible configuration. For more information on the class 5 PKI system, log onto the DISA website at www.disa.mil.

VI. PRACTICAL APPLICATION

A. PROBLEMS WITH THE PKI SYSTEM

It is very difficult to determine someone's identity on the Internet. Even the use of a digital certificate does not guarantee a binding between a public key and a person's identity. The current specification of the DoD PKI demands a great deal of trust from the user.

The value of a certificate is limited by a user's ability to verify the CA's digital signature on the certificate. The user must obtain an authentic copy of the CA's public key to verify the digital signature. In current systems, the major commercial PKI vendors have given their public keys to companies, for example Microsoft and Netscape, so these companies can include their public key in their browser systems. However, if a user receives a certificate from a CA that is not preloaded into a browser, the browser will warn the user that the CA is not recognized. The browser will then ask whether the user is willing to accept the CA's public key. The user has to make the decision without any knowledge concerning the CA. in order to use the certificate, the user must place full trust in an unknown CA. (Hombeck, R., 1997)

The user also places trust in the browser. A malicious entity can place counterfeit public keys in a browser that is locally accessed (e.g., computers in a public library, or coffee shop). The malicious entity can preset the default web site to access a "spoofed web site." If the user wants to initiate a secure transmission, the browser will accept the spoofed site's certificate.

Under the current PKI schema, the user has to trust the CA to properly authenticate an applicant's identity. If the CA does a poor job of authentication, then a valid binding of the public key to the individual cannot occur. In fact, the user has no fail-proof way of determining whether the CA is properly authenticating individuals. Even if the user can obtain a policy statement from the CA, there is no guarantee that the company is abiding by the policy. In many cases the policy is written by attorneys that use vague language to ensure that the CAs are not liable in the event that they fail to

perform some task. The user may be able to determine the effort the CA exercises in authenticating individuals by a class system or through the cost of the certificate, but this is not a good measure of the practices actually used. The user may also collaborate with other users to test the CA by masquerading as someone else. If the CA detects the deception, then the users will have more confidence in their authentication policies.

A CA may have difficulty accurately identifying an individual. The ease for a malicious entity to assume another person's identity depends on the level of authentication used. Identity theft is easy to perform and is difficult to detect. In some cases, all that is needed is a person's social security number and name. Once that information is obtained, the malicious entity can obtain fake drivers licenses, passports and can apply for credit cards, under the real person's name. Unless extensive authentication is performed, a CA will not be able to catch someone using identity fraud. The user has to trust the CA's belief that the individual has been properly authenticated. Unfortunately, the user has no method of evaluating the CA's decision.

The use of trust models can help to answer some of the questions that the current PKI systems cannot. Implementation of trust models may allow the user to obtain the information necessary to make an educated decision on whether a certificate should be trusted, or not.

B. INCORPORATING TRUST MODELS IN THE DOD PKI SYSTEM

In this section we use Audun Josang's trust model with the DoD PKI model to evaluate the steps necessary to develop a practical model. The Josang model was chosen because it is the most comprehensive and flexible model that we reviewed.

Both the Josang trust model and the DoD PKI model were developed in an effort to establish the identity of an individual over the Internet. Recall that in Josang's trust model, to determine the authenticity of a certificate, the user had to form three opinions. The first opinion dealt with the authenticity of the public key belonging to the CA. This is an evaluation of the key-to-CA binding. The second opinion was in the recommender ability or verification ability of the CA. The last opinion concerned the CA's actual recommendation of the certified key, or a verification of the key-to-name binding.

To use Josang's terminology, agent A's opinion in the key authenticity (KA) of certificate authority B's public key is represented as $W_{AKA(kb)}$. Agent A must also form an opinion on recommender trustworthiness (RT), which measures A's trust in B to properly certify keys. Agent A's opinion about B's ability to certify keys, is represented as $W_{ART(B)}$. Agent A must also evaluate the recommendation of B as to the validity of agent C's public key or the identity of agent C. Certificate authority B's opinion is represented as $W_{BKA(kc)}$.

In order to validate the authenticity of the certificate, A must first evaluate the recommendation from certificate authority B. Agent A will combine its opinion of B's key authentication with its opinion about B's recommender trustworthiness. This will determine agent A's opinion about B's capability as a recommender. The recommendation will be represented as the conjunction of both opinions $W_{AB} = W_{ART(B)} \wedge W_{AKA(kb)}$.

Agent A must then combine its opinion about B's recommendation ability with B's recommendation about C's public key. Agent A's opinion about agent C's public key based on B's recommendation will be represented as $W_{AKAB(kc)} = W_{AB} \otimes W_{BKA(kc)}$. (Josang, A., 1998)

The DoD model is built upon a hierarchical architecture. However, in order to be interoperable with other PKI systems within the government and the commercial sector, cross certification or the use of a third party CA is needed. As a result, we will apply the Josang trust model to a system that has a hierarchical architecture, but uses cross certificates.

An additional assumption is that the PKI system will have a program, similar to REFEREE (Chu, Y., and others, 1997), that will be able to extract information from the X.509 system, gather input from the user via a GUI interface, compute the trust algorithm, and provide an output that will assist the user in determining a level of trust in a proposition x. The actual programming and design of the software program is beyond the scope of this thesis.

C. CA'S OPINIONS ON KEY AUTHENTICITY

The CA's opinion about the identity of an individual can be included in an extension of the X.509 certificate. The DoD PKI system uses local registration authorities (RAs) to verify the identity of individual service members. The DoD literature did not specify the methods of authentication, but it makes sense to assume that the RA will require different levels of authentication depending upon the security clearance of the individual and the attributes that are assigned. If we assume three levels of authentication, corresponding to Confidential, Secret, and Top Secret clearance, then the method of authentication should become more rigorous accordingly.

Currently in the DoD, if a Top Secret meeting is called, the personnel invited to the meeting must have their security office send a message to the group organizing the meeting stating that they have the appropriate security clearance to attend the meeting. When the person shows up at the meeting, his or her identification is verified by checking their government-issued identification card, comparing their name against the list of attendees, and by verifying that their command passed their security clearances for them. In the PKI system, the CA replaces the identification card and depending on the type of certificate, the security office by issuing a certificate with security-level attributes. When an individual arrives at a meeting, or joins an electronic meeting, he can be authenticated by his certificate, which may or may not include the level of his security clearance. The DoD PKI must support cooperative work environments, such as video teleconferencing and electronic conferences with commercial industries. If the PKI certificate is going to be used for authentication, then it is necessary to some degree for the user to have confidence in the CA's identification and authentication procedures.

In the DoD PKI system, authorization of a Confidential certificate may simply invoke checking the member's identification card. Authorization of a Secret certificate may require a unit's security officer to verify the member's security level, authenticate him against his smart card, and accompany the individual to register with the RA. The Top Secret authentication may require fingerprint checks in addition to the requirements for the Secret certificate.

This methodology may work sufficiently well within the DoD system, but what happens when cross certificates are issued? How will CAs outside of the DoD domain authenticate individuals? When the certificates are issued, the CA will list its belief, disbelief, and uncertainty in the authentication of the individual listed on the certificate in one of the extensions of the certificate. Due to the extensive authentication measures used with a Top Secret certificate, the CA's opinion might be (.95, 0, .05). A Confidential certificate might have an opinion of (.85, .05, .10).

Although the opinion value that a CA assigns to a certificate is subjective, a scale called a subjective index can be used. A standardized subjective index will correspond to the level of the authentication performed by the CA. For example if a certificate is issued based on an e-mail address, the opinion could be (.4, .3, .3). If an individual verified himself to the RA using a driver's license, the opinion value could be (.6, .1, .3). The subjective index would be broad enough to allow the CA some flexibility, but not enough to allow a CA to assign arbitrary values. Each CA, and the applications that the certificate will be used with, will determine the level of granularity assigned to the values.

When a certificate is built, the CA's opinion about the validity of the certified key can be incorporated into the certificate using the extension fields specified in the X.509 v3 standard. Anyone receiving the certificate will be able to utilize the CA's opinion to determine his or her trust in the key-to-owner binding.

D. RECOMMENDATION TRUSTWORTHINESS

Under this system, every CA will include its opinion concerning the key-to-owner binding in the certificate. The opinion will be based on the level of authentication performed. When a user receives a certificate, he can begin to formulate an opinion about the validity of the certificate, but the user must also have an opinion about the trustworthiness of the CA's recommendation.

If a user has little faith in the recommendation trustworthiness (RT) of the CA, then the certificate that it issues is worthless to the user. How can a user determine if a CA is properly authenticating individuals, or is researching the CAs with which it has

cross certifications? There are a couple of ways that an individual can gain the necessary information upon which to form an opinion.

The easiest way in which to form an opinion on the recommendation ability of a CA is to base it on personal experience. When a person registers with an RA or CA for a certificate, he or she can determine the verification methods, the organization's policies, and the professionalism of the organization first hand. By making an assumption that the RA or CA always acts in a similar method, the person can form an opinion about its recommendation ability. An RA or CA may receive high-recommendation opinions if it can show that it is competent (the staff is knowledgeable), thorough (the staff follows all required verification procedures), and professional (the staff has the skills and desire to perform their tasks in a business-like manner.)

The obvious weakness of this method is that the evaluation of a particular RA may not be indicative of the other RAs in a domain. Additionally, to form an educated opinion, an individual must understand the RA's policies and know what procedures must be followed to assess a CA with a high degree of accuracy. Since there is no standardized method of analysis, an informal risk assessment must be made by each user. This means that the user must assign weighting factors to the attributes he or she feels are important. This can be a very laborious and time consuming process. The results may vary widely as some users will do a better job of assessing a CA than others. Most people are not willing to spend the time or the effort to properly evaluate a CA or RA. Additionally, the user may not be informed enough to make an educated opinion based on first-hand knowledge. A CA may appear extremely competent, but the CA may be violating numerous security precautions.

Another method to gain information on a CA's recommendation competence is to ask trusted friends or colleagues. The consensus operator in subjective logic ($\oplus$) allows a user to combine the opinions of multiple individuals. Consensus operators are supposed to reduce uncertainty. In this scenario, friends A and B will provide their opinions about a CA's recommendation trustworthiness (RT). If $W_{A(RT)}$ represents A's opinions about a CA's recommendation trustworthiness, and $W_{B(RT)}$ represents B's opinions about the

same CA, then $W_{A(RT),\, B(RT)}$ represents the consensus opinion. The consensus operator uses the symbol '$\oplus$', so $W_{A(RT),B(RT)} = W_{A(RT)} \oplus W_{B(RT)}$.

This method is based on three important assumptions. The first assumption is that the friends or colleagues have obtained their information first hand, and that they have enough knowledge to make an informed decision regarding a CA's RT.

The second is that the colleagues are trusted to make recommendations. How much does the user trust his colleagues? Do they have unconditional recommendation trust? To derive an opinion, the user's opinion concerning his colleague's recommendation trustworthiness will have to be combined with his colleague's opinions concerning a CA's recommendation trust.

The third is that the user either obtained the information from colleagues verbally, or through an "out of band" method. If colleagues transmitted the recommendation over the Internet, using a digital signature, it is assumed that the user already has confidence in the colleague's certificate. If not, then the user should not use the recommendation, as he has not taken the steps necessary to develop an opinion on the certificate's key-to-identity binding.

For example, the F-18 program manager meets with a new contractor that manufactures environmentally friendly aviation paint. During the meeting, the program manager and the contractor exchange certificates. The next day the program manager wants the supply department to develop a request for proposal and send it the contractor. When the contractor signs the request for proposal with a digital signature, and returns it to the supply department, the supply department must obtain the contractor's public key for verification of the signature. Since the program manager already has the contractor's public key, he can send the supply department the contractor's public key, signed with his own key. The supply department can verify the program manager's digital signature, which ensures the message's integrity and origin and then use it to verify the digital signature.

If the user relies on information from colleagues, the user must first form an opinion about his colleague's RT. For every colleague or trusted source that is providing

an opinion about the recommendation trustworthiness of the CA, the user must individually form an opinion about the source's RT. That opinion will be combined with the source's opinion of a CA's RT using the subjective logic operator called recommendation. For each friend the operator will have the same form.

U is the user's opinion. The user's opinion about B's recommendation trustworthiness is represented as $W_{URT(B)}$. B's opinion about the CA's RT is represented as the proposition 'x', so the opinion will be represented as W_{Bx}. The symbol '$\otimes$' is used to designate a recommendation. U's opinion about the CA's RT, based on B's recommendation, is represented as $W_{URT(B),Bx} = W_{URT(B)} \otimes W_{Bx}$.

Once a user has determined his opinions of each individual's recommendation on the CA's RT, the opinions can be combined. The consensus rule in subjective logic is used to combine independent opinions into a single combined opinion. The consensus operator combining the opinions of the user's friends A and B would be represented as $(W_{URT(A)} \otimes W_{Ax}) \oplus (W_{URT(B)} \otimes W_{Bx})$.

If this method were implemented, the PKI system would have to be able to extract the CA's opinion concerning the authenticity of the individual from the X.509 certificate, it has to receive and process the colleague's opinions on the CA's RT, and it must gather information on the user's opinion on his friend's RT. Current programs can extract information from X.509 certificate extension fields, but the colleague's opinions must be manually entered into the program, appended to the certificate, or entered into the certificate in a recommendation extension field.

If the latter option is used, a system similar to the PGP method of signing certificates can be used. Once a certificate has been modified with an entry in the recommendation extension field, or appended, the person modifying the certificate will take a hash of the entry and sign it with his or her private key. The user's computer program would store the certificate in a public key ring and have an owner trust field (user's opinion regarding the RT of that key or friend) and a CA RT field (extracted from the certificate). The values stored in these fields would be in the format of Josang's model (belief, disbelief, and uncertainty). The program will use those values to perform

recommendation and consensus operations. Instead of using a key-legitimacy field, the program will output a value that will assist the user in determining trust in a key-to-owner binding.

This method would also require the use of a subjective index to evaluate both the values that the friends assign to the CA's RT and the value that the user assigns to his friends' RT. The index is used to provide guidance to the friends and the user when assigning subjective values to their opinions. The index also provides a measure of standardization. An example of a subjective index used to assign values to a colleague's RT might be as follows:

- Very familiar with colleague's RT abilities – (belief value from 1.00 to .90)
- Somewhat familiar with colleague's RT abilities – (belief value from .89 to .80)
- Not very familiar with colleague's RT abilities – (belief value from .79 to .60)
- Not familiar with colleague's RT abilities – (belief value from .59 to .40)
- Usually do not trust colleague's RT abilities – (belief value from .39 to .20)
- Never trust colleague's RT ability – (belief value from .19 to .0)

Another method that can be used to gain information about a CA's recommendation competence is to rely on the results published by an outside auditing organization. In the DoD PKI system, RAs are used to register individuals. The RAs abide by a central policy that is promulgated from the root CA. Unfortunately, it is difficult for a user to determine if the RA is actually abiding by the policy. Within the DoD, inspections of the RAs are performed by organizations within and outside of the DoD to ensure that policies are being followed. Unfortunately, inspections in and of themselves do not ensure that the policies will be followed, especially if there are no repercussions for a failed inspection. The inspections do, however, provide guidance to the user. If a particular RA is continually receiving poor scores on inspections, then the user may assign them a recommendation opinion of (.4, .4, .2). The agency can post their opinions of a CA's RT on a public web page that can be manually entered into the PKI system, or they can add their RT opinion in the CA's certificate along with a hash.

However, the user must again form an opinion on the recommendation trustworthiness of the certifying/inspection agency. This opinion will have to be factored into a recommendation or consensus opinion (if more than one agency inspected the CA). This method assumes that an agency exists that can and will inspect or audit CAs to ensure that they are complying with accepted standards and public laws. Users can use the results of the inspections to form opinions about the RT of the CA.

At this point we have made two assumptions. The first is that the results of the inspections are public and accessible. If they are not, then the individual has to rely upon the opinions of other sources. The second is that the user has a trusted copy of the inspector's public key to verify the hash. The public key could be obtained from the agency, or through the military's Communication Security Material System (CMS).

The current policy within the DoD is that any CA that is cross certified would have to be certified by NSA or an NSA approved independent auditor. NSA would ensure that the CA being certified meets its minimum accepted criteria (although that criteria is not defined at this time). The thoroughness of the inspection will depend on the security level of the information that the certified CA processes. NSA could provide the cross certification certificates as well as providing their opinion of the RT of the CA in the recommendation trust field in the X.509 certificate.

The agency certifying the CA does not have to be a third party; rather, it may be a domain's root CA. When a cross certification certificate is issued, it may contain the root CA's opinion concerning the RT of the certified CA. If due to periodic inspections, the root CA determines that it is necessary to revise its opinion of the RT of the certified CA, then it will issue a new cross certification certificate. The amount of trust that a user or outside auditing agency places in the CA's recommendation trust will be based in part on the frequency with which it monitors its own domain's CAs and the other CAs with which it is cross certifying.

E. TRUST IN THE CERTIFYING KEY

The final opinion that a user needs to formulate is the user's trust in the private key used to certify the certificate. When a certificate is made, it is digitally signed by the

manufacturing CA. To verify the signature, the user must have a trusted copy of the CA's public key. However, how does a user obtain a trusted copy of the CA's public key?

There are a number of ways to obtain a copy of the CA's public key. Some methods are more secure than others. The methods used are a trade off between certainty, cost, timeliness (of response and data), and risk.

The first method is to obtain a copy of the public key from the RA or the CA during the registration process. Utilizing a manual exchange, the RA can distribute the CA's public key on a floppy disk.

A physical exchange of this type is the most secure method of obtaining a public key. If the key is fraudulent, the user immediately knows the source of the problem. As there were no middlemen involved in the distribution of the key, the user can hold the CA or RA directly responsible for distributing a compromised or fraudulent public key. However, just because a key is physically received from the CA, does not mean that a user should place full trust in the key. The user should also consider factors such as the following: the physical security of the RA site, the reliability of the staff, and the method that the key is written to the floppy (was the key generated by the CA, or was it downloaded from a computer at the RA's office). If a subjective index of trust was used, a belief value for a public key obtained through this method could be between 1.0 and .95, because this is one of the most secure ways of obtaining a public key.

Another method that is available within the DoD system is to use the military's CMS system to distribute the CA's public key. This is the same method that is used to physically transfer top-secret material and symmetric keys. It has proven itself to be highly reliable under heavy scrutiny. The public key could be delivered to a unit's security officer who could then distribute it to personnel in the unit needing the CA's public key. This is the most secure method available to deliver the CA's public key to users that do not have direct access to the CA or RA. If this method were utilized, a user's confidence in the CA's public key could be between 1.0 and .90, because this is also a very secure method of obtaining the CA's public key.

Under the CMS system, the user would have to form an opinion about the reliability of the distribution method, the possibility that the public key may have been compromised before distribution, and the belief in the reliability of the security officer or CMS custodian.

Utilizing electronic methods, one can also obtain the CA's public key from the CA's directory. The user then has to decide if the public key is legitimate. It is possible for a malicious entity to reprogram a DNS server to advertise a false IP address. The entity can then build a directory similar to the CA's actual directory and place forged public keys in the directory. Depending upon the security policies of the CA, it may also be possible for a malicious entity to place a forged public key on the CA's actual directory. When this method is used, it is difficult to determine if the CA's public key is legitimate.

A more secure method is for the RA to issue a session key to the user during registration. The RA then securely transmits the same session key and identification information to the CA. The user then contacts the CA, authenticates, and uses the session key to request the CA's public key. The CA then sends the public key to the user encrypted with the session key. The user can then determine authenticity because only the RA, CA, and the user know the session key. This method can be more secure by adding nonces and time stamps to foil man-in-the-middle attacks.

In the DoD PKI system, the user must be able to enter his opinion on the trustworthiness of the CA's public key into the system to determine the total trust in the certificate. Using a GUI interface, the PKI system could obtain the user's opinion on the trustworthiness of the CA's public key. If the key was distributed using the CMS system, then the opinion might have the value (.95, .025, .025). If the key was obtained from a public bulletin board, then the opinion might have the value (.65, .20, .15).

F. TRUST IN A CERTIFICATE

In order to determine trust in the DoD PKI system, the user must form an opinion on his trust in the CA's public key and combine that opinion with his trust in the CA's recommendation trustworthiness. Those opinions must then be combined with the CA's

trust in the authenticity of the individual being assigned a certificate. The CA's belief in the individual's identity will be included in one of the certificate's extension fields when it is issued.

The first opinion that must be formed is the user's trust in the public key of the CA that has digitally signed the certificate. The trust values assigned will be dependent on the method of key distribution. The most effective methods are physical distribution of the key utilizing the CMS system or obtaining the key from a CA or RA. The least preferred method is to obtain the certificate from a bulletin board or third party.

In the DoD system, the trust model will not include an assumption that the user has enough knowledge about the procedures and business practices of the CA to form an educated opinion about the trustworthiness of the CA's recommendations. Instead, the recommendation trustworthiness of the CA will be obtained from third party organizations that either certify or inspect the CA. The ratings or evaluation marks will be used by the user to form an opinion on the CA's recommendation trustworthiness. These opinions will be combined with the user's opinions on the trustworthiness of the third party organization to properly conduct an evaluation. We are assuming that the user will not want another organization to evaluate the independent inspector, as the inspectors will have to be reputable (e.g., U.S. General Accounting Office).

The recommendations from the independent inspector will have to be combined with the user's belief in the recommendation trustworthiness of the inspection agency. Again we are assuming that the inspection agency is reputable, and independent, so the user's belief opinion of the inspectors recommendation trustworthiness could be between 1.0 and .90. Another outside agency inspecting the third party inspectors is not necessary to form an opinion on the trustworthiness of the third party organization.

The final opinion to be evaluated is the CA's opinion on the authenticity of the identity of the individual receiving the key. The trust opinion value assigned will depend on the method of verification. The trust value will be placed in an extension field of the certificate.

If B represents the CA and U represents the user, then U's opinion on the key authenticity of the CA is represented as $W_{UKA(kb)}$. KA represents the key authenticity of the public key 'kb.'

User U's opinion about a CA's recommendation trustworthiness, based on the GAO's report (W_{GAOx}) is represented as a recommendation in subjective logic as $W_{URT(GAO),GAOx}$. $W_{URT(GAO)}$ represents U's opinion on the recommendation trustworthiness of agency GAO. W_{GAOx} represents the GAO's trust in the proposition x, which in this case is the recommendation trustworthiness of the CA. $W_{URT(GAO),GAOx} = W_{URT(GAO)} \otimes W_{GAOx}$. If the user does not have complete trust in the public certificate of the GAO, then he can combine his opinion on the authenticity of the GAO's public key into the formula. It would have the following format: $W_{UGAO} = (W_{URT(GAO)} \wedge W_{UKA(kgao)}) \otimes W_{GAOx}$. We will make the assumption that the user has complete trust in the GAO's public key to simplify the formulas, however, the user's opinion on the validity of the GAO's certificate can be easily incorporated into the trust model.

The user's opinion on the recommendation trustworthiness of the GAO will be combined with U's opinion on the authenticity of B's public key in the form of a conjunction. U's opinion on B's ability to make a recommendation will have the following form: $W_{UB} = (W_{URT(GAO)} \otimes W_{GAOx}) \wedge W_{UKA(kb)}$.

This opinion is combined with the CA's trust opinion on the authenticity of the individual receiving the certificate. If C represents the individual receiving the certificate, then B's opinion on the key authenticity of C will have the form $W_{BKA(kc)}$. For U to form a trust opinion based on B's opinion, he must combine B's opinion with his opinion regarding B's ability to recommend. So using the previous formula, U's opinion about C's certificate will be represented as $W_{UKAB(kc)} = ((W_{URT(GAO)} \otimes W_{GAOx}) \wedge W_{UKA(kb)}) \otimes W_{BKA(kc)}$.

An example illustrating the subjective logic is as follows: U's opinion on the recommendation trustworthiness of the GAO is (.90, .05, .05, .50), representing belief, disbelief, uncertainty, and atomicity. Atomicity is .5 because the user either believes or disbelieves the GAO's recommendation trustworthiness. The GAO's opinion on the

recommendation trustworthiness of B is (.95, .025, .025, .5). Using the formula $W_{URT(GAO)} \otimes W_{GAOx}$ yields a resulting opinion of (.85, .023, .122, .5).

- $b_{UGAOx} = b_{URT(GAO)} b_{GAOx} = (.90)(.95) = .85$
- $d_{UGAOx} = b_{URT(GAO)} d_{GAOx} = (.90)(.025) = .023$
- $u_{UGAOx} = d_{URT(GAO)} + u_{URT(GAO)} + b_{AB} u_{GAOx} = .05 + .05 + (.90)(.025) = .122$
- $a_{UGAOx} = a_{GAOx} = .50$

This value must be combined with U's belief in the key authenticity of B's public key, which will be (.90, .025, .025, .5). Using the formula $(W_{URT(GAO)} \otimes W_{GAOx}) \wedge W_{UKA(kb)}$ the resulting opinion is (.77, .05, .18, .49). In the formulas below, 'x' represents U's opinion on the B's recommendation trustworthiness and 'y' represents U's opinion on the authenticity of B's public key.

- $b_{x \wedge y} = b_x b_y = (.86)(.90) = .77$
- $d_{x \wedge y} = d_x + d_y - d_x d_y = .023 + .025 + (.023)(.025) = .05$
- $u_{x \wedge y} = b_x u_y + u_x b_y + u_x u_y = (.85)(.075) + (.122)(.90) + (.122)(.075) = .18$
- $a_{x \wedge y} = (b_x u_y a_y + u_x b_y a_x + u_x a_x u_y a_y) / (b_x u_y + u_x b_y + u_x u_y) = (.85)(.075)(.50) + (.122)(.90)(.50) + (.122)(.50)(.075)(.50) / (.85)(.075) + (.122)(.90) + (.122)(.075) = .49$

The resulting opinion value represents U's opinion on B's recommendation ability. To determine U's trust opinion on agent C's certificate, the opinion value just computed must be combined with the value that B assigned to his opinion on the authentication of agent C, in this case (.95, .025, .025, .5). Using the formula $((W_{URT(GAO)} \otimes W_{GAOx}) \wedge W_{UKA(kb)}) \otimes W_{BKA(kc)}$ yields a final trust opinion value of (.73, .019, .25, .5). In the formulas below, 'x' represents U's opinion on B's recommendation ability and 'y' represents B's opinion on the authenticity of C's identity.

- $b_{xy} = b_x b_y = (.77)(.95) = .73$
- $d_{xy} = b_x d_y = (.77)(.025) = .019$
- $u_{xy} = d_x + u_x + b_{xy} = .047 + .18 + (.77)(.025) = .25$
- $a_{xy} = a_y = .50$

Once U has calculated his trust opinion, he can make the decision to trust or not to trust the certificate. The trust decision is based upon the opinion calculated, but it is still a subjective decision that will differ among different people. A subjective index could be used as a guide, but the final decision can depend a number of factors including a person's risk aversion, or the value of the transaction. In automated systems, such as those used with intelligent agents, the decision to reject or accept trust will be based on predefined threshold values established by a policy-level decision, which in turn are implemented as procedures in software and hardware.

G. CHAINING TRUST

Determining a CA's public key becomes more difficult when evaluating CAs outside of the DoD domain. The standard method for obtaining the public key from a CA that is outside of the DoD domain is through cross certification. The DoD root CA will issue a cross certificate that contains the public key of the CA that is outside of the domain. This certificate is signed by the root CA to verify its authenticity and integrity. The user obtains the cross certificate from a bulletin board, from the CA, or from its domain CA, and verifies the certificate by decrypting the hash with the root CA's public key. The user can then verify any of the certificates in the other domain by comparing the signatures on the other domain's certificates against that CA's public key.

When evaluating a cross certificate, the user must first form an opinion on the domain CA's ability to make a recommendation by evaluating the CA's public key and recommendation trustworthiness. The opinion will have the same format as previously mentioned: $W_{UB} = (W_{URT(GAO)} \otimes W_{GAOx}) \wedge W_{UKA(kb)}$.

The primary purpose behind cross certification is to validate the public key of the CA that is being certified. However, as discussed earlier, knowing the public key of a CA is only part of the trust equation. The user also needs information on the CA's recommendation trustworthiness. The CA is responsible for thoroughly evaluating a CA that is going to be cross certified. The CA being certified must meet certain minimum acceptable standards (which still need to be identified). Based on the CA's evaluation of the CA being certified, it will form an opinion on the certified CA's recommendation

trustworthiness and the validity of its public key. If B represents the CA and D represents the CA being certified, then B's opinion on the recommendation ability of D is represented as $W_{BD} = W_{BRT(D)} \wedge W_{BKA(kd)}$.

The user's opinion about CA D's ability as a recommender will combine U's opinions about B's ability as a recommender with B's opinion on D's ability as a recommender. It can be expressed by the formula $W_{UB} \otimes W_{BD}$. This formula can be expanded to include other cross certified CAs. If E represents another CA that is cross certified by D, then the opinion can be expressed as $W_{UB} \otimes W_{BD} \otimes (W_{DRT(E)} \wedge W_{DKA(ke)})$.

If a third party organization such as the GAO is used to obtain information about a CA's recommendation trustworthiness, then the user would have to combine that information with the CA's opinion on the validity of CA D's public key. U's opinion on the validity of D's public key based on the assessment by CA B would be $W_{UB} \otimes W_{BKA(kd)}$. User U's opinion on the recommendation trustworthiness of D based on GAO's assessment is $W_{URT(GAO)} \otimes W_{GAOx}$, where 'x' represents D's RT. The formula to determine recommendation ability of D, based on B's recommendation on the public key and the GAO's recommendation on D's RT is as follows: $W_{UD} = (W_{UB} \otimes W_{BKA(kd)}) \wedge (W_{URT(GAO)} \otimes W_{GAOx})$.

The Audun Josang model can be implemented within the DoD PKI system. The model can address the three areas in a PKI system that require trust determinations. The model combines a user's opinion on the CA's public key with the user's opinion on the recommendation trust of the CA. It then combines these opinions with the CA's opinion on the authenticity of the identity of the entity applying for a certificate. However, the practical implementation of this model with the DoD PKI systems will require significant managerial effort and pre-planning. The implementation can succeed, but it will require a great deal of effort, cooperation, and coordination in resolving issues such as the following: interoperability issues with other agencies and the commercial sector, standardizing the trust model's subjective index, standardizing the CA's authentication procedures, developing an audit check-off list for third party agencies to use when inspecting CAs, determining how the CA's public key will be distributed, agreeing on the

programming language that will implement the trust model, and determining how to incorporate the trust model in the PKI system (e.g., key rings, third party auditors, software requirements, interfacing with the X.509 standards).

VII. DECISION-MAKING STRATEGIES

A. TRUST MODELS AS DECISION MAKING TOOLS

Given that trust models are based upon subjective inputs to the decision-making process, can these models be incorporated into formal analysis and be practically applied? There are arguments against devising models that try to quantify the unquantifiable. The question is: what is quantifiable? In a military context, can a modeler a priori gauge performance of humans in combat conditions? A horse trainer may not be able to give an objective formula for ranking a winning horse, but he may be able to rank order one horse over another. The trainer can readily evaluate the horse's stamina, his stride, the shape of the head, and overall look of the horse, but assigning values to aggressiveness, courage, and winning spirit is more difficult. In order to rank a horse, the trainer has to quantify some of the horse's attributes through subjective introspection instead of an objective formula. Assigning objective scales to subjective attributes can be done, however, by quantifying subjective attributes using expert opinion and accumulated experience. (Keeney, R., and Raiffa, H., 1993)

This presents a problem. If an expert is not available and a layman must convert a subjective attribute into an objective scale, can the information be used in a formal model? Does the individual have enough knowledge of the problem to be able to assign a value to the subjective attribute? Is the individual willing to take the time to research the problem before assigning a metric? Can the individual make an objective determination of a scale? How will human stress affect the results? How valid are conversions from subjective attributes to objective ones?

B. DECISION MAKING

Solving complex problems requires a formal strategy. There are numerous ways to approach problem solving. Traditional approaches usually consist of five steps. The first is to define the context, or identify the underlying problem that must be dealt with. Next, a list of alternative actions or inactions is generated to deal with the problem. The consequences of each alternative are then determined. In this phase, probability (or

likelihood) expectations for various courses of action can also be generated. The alternatives must then be evaluated. The criteria used for evaluation and methods to resolve tradeoffs between different combinations of objectives must be determined in this phase. The final step is choosing an alternative. (Stokey, E. and Zeckhauser, R., 1978) Other strategies include assigning attributes to the alternatives and then developing mathematical models to test the alternatives. (Render, B. and Stair, R., 1997)

Another approach uses value-focused thinking. A value is a principle (e.g., ethical, risk aversion, priority) that is used for evaluation. They can be used to evaluate action, inaction, alternatives, and decisions. This approach starts by asking what the decision maker hopes to achieve in the decision context. The values indicate what information is important. Instead of listing alternatives, the decision makers list the values that they would like to achieve. These values are transformed into objectives, which are then evaluated for completeness and importance. Alternatives are then generated in terms of achieving the specified objectives. Next, alternatives are evaluated by measuring attributes, which represent the extent to which an objective has been achieved. The final step is to select the best alternative. (Keeney, R., 1992)

C. FORMAL MODELS

If the decision maker is the only person to whom the decision applies, it is reasonable to use subjective values for the various attributes. He can use whichever attributes he feels are complete. He is also free to analyze the problem to whatever degree he feels comfortable. The fact that the decision does not have to be explained or rationalized gives the decision maker considerable flexibility. This is similar to the trust model on which Pretty Good Privacy protocol (PGP) is based: the user is free to use any method to determine a level of trust in the key owners, because the user is the only person using that information.

When communicating how a decision was made, the decision maker has to explain his rational. This means he must be more formal in his approach. He will have to explain the objective hierarchy in greater detail (i.e., decompose the problem). Additionally, the attributes that are assigned should be objective instead of subjective

whenever possible. The decision maker may also have to analyze the problem in more depth (e.g., explain all of the reasonable components of the problem). This allows another person to objectively view the attributes and make a rational decision using the given information. If there is not enough information, or if the values of the attributes are subjective, then someone other than the original decision maker may have more difficulty arriving at a conclusion.

In complex problems, solution methods dissect the problem into pieces and assign values to each piece. In cost-effectiveness analysis, the results of a particular course of action in response to a given problem are broken down into costs (C) and n benefit measures ($B_1,\ldots,B_n$). The action is described in terms of ($C, B_1, B_2,\ldots,B_n$). The benefits have different measurement metrics assigned to them, so they cannot be combined into one composite benefit measure.

The decision maker usually has a fixed cost as a constraint, so he will narrow his courses of action by staying within the fiscal constraint specified by (C). In addition he will try to maximize the benefits, but this can be very difficult when all of the benefits have incompatible measurement units.

To compare one alternative to another, cost-effectiveness analysis preassigns aspiration levels to each benefit measure ($B_1, B_2,\ldots,B_n$). The alternatives that can meet or exceed the aspiration levels are the best choices. If none of the alternatives meet the levels, the levels can be lowered until an alternative is selected.

Unfortunately, this method does not explain how aspiration levels should be selected. It also does not explain how tradeoffs among the benefits will be performed. Uncertainty and ignorance are also not factored into the method.

Another method used to solve complex problems is cost-benefit analysis. A cost-benefit analysis is similar to a cost-effectiveness analysis, except that the benefits of a particular course of action are all combined into a single measure. Conversion factors are applied to the benefits to change them into like units of measure. If B_0 is the composite benefit value and w1 is the conversion factor assigned to the first benefit, the formula would be $B_0 = w_1b_1 + w_2b_2 + w_3b_3\ldots w_nb_n$. (Keeney, R. and Raiffa, H., 1993)

Each alternative can now be compared using the same measure. In the cost-benefit analysis, each alternative will have a cost associated with it. The composite benefit value divided by the cost will give a benefit-to-cost ratio. Using this ratio, the various alternatives can be evaluated. This method is easier to evaluate than the cost-effectiveness analysis, because the decision maker does not have to perform any tradeoffs.

The difficulty that arises in a cost-benefit analysis is that of determining a suitable conversion factor. These conversion factors are generally assigned using subjective valuations. In cost-benefit analyses, benefits are typically converted into monetary units, but the determination of worth is still a subjective decision. In practice, some important determinates in decision making such as psychological, political or security are not included in a model because a suitable market mechanism does not exist to price a particular benefit. In addition, this technique also does not account for ignorance or uncertainty.

D. ATTRIBUTES

An attribute provides a scale for measuring the degree to which an objective is met. To be useful, an attribute must have two properties: comprehensiveness and measurability. An attribute is comprehensive if by knowing the measurement of an attribute the decision maker can understand whether its associated objective is achieved. Comprehensiveness refers to the extent of information provided and whether it meets the decision maker's needs.

An attribute is measurable if it is possible to obtain a probability distribution for each alternative over the possible levels of the attribute and if the decision maker's preferences for different attribute levels can be assessed. In other words, is it possible to get the necessary assessments and can the attributes be weighed or ranked by the decision maker? (Keeney, R. and Raiffa, H., 1993)

Keeney and Raiffa also specified five properties that are desirable in a set of attributes. The first is completeness. A set of attributes is complete if it can adequately

cover all of the important aspects of the problem. The attributes must indicate to what extent or degree that an objective is met.

Attributes must also be operational. The attributes must be meaningful, so the decision maker can use them to analyze alternatives. Additionally the attributes should be able to support a particular position when explaining a decision to others.

The set of attributes should be decomposable. If a set of five attributes are used to evaluate an objective, the attributes should be able to be decomposed to the sub objectives that they apply to. Three attributes may describe sub-objective X_1, while the other two apply to sub-objective X_2.

A set of attributes should not be redundant. This seems obvious, but often one attribute may only have significance when described in terms of another attribute. An example is a missile shot where the attributes velocity and Doppler effect are used to describe the shot. In this case the attributes are redundant because velocity is used in the calculation of the Doppler effect. However, it may be necessary or expedient to use redundancy for optimization of a computation or other aspects of processing. Another common error with a set of attributes is that some of the attributes apply to the input to a system, and others apply to the output.

The last property is that the set of attributes should be kept to a minimum. As the number of attributes increases, determining joint probability distributions and quantifying multiattribute preferences becomes more difficult. (Keeney, R. and Raiffa, H., 1993)

E. SUBJECTIVE ATTRIBUTES

There are three ways to deal with a subjective problem. The first is to develop a subjective index that measures the objective. For each alternative course of action, probability distributions are assigned to describe the impact in terms of the index, and a utility function is assigned to the attribute. Expected conditional utility can then be computed. Another method is to use a proxy attribute. This is done the same way as a subjective index, but the utility function is assigned to the proxy attribute. The third way is a direct preference measure. This method directly assigns a conditional probability utility to the alternatives of a subjective problem. This method bypasses assigning

attributes to the problem, but requires that the decision maker have more knowledge of the problem than the other methods. It assumes that the decision maker understands the relationship between the problem, the alternative solutions, and the attributes involved. (Keeney, R. and Raiffa, H., 1993)

Proxy attributes attempt to assign objective attributes to a subjective objective. A proxy attribute does not directly measure an objective, but can be used to describe the degree to which an objective has been met. It indirectly measures an objective. Rather than explain the Bayesian theory and probability distributions used, we use an example to illustrate the concept.

In modeling trust, if the objective is to determine whether Tom is trustworthy, the set of attributes used to determine the objective are largely subjective, because the concept of trust is subjective. Attributes to apply to trust would be reputation, belief that he is trustworthy, disbelief that he is trustworthy, ignorance, and uncertainty. Proxy attributes might consist of the following: Does Tom have a prison record? What is Tom's financial position? Does Tom have any financial problems? What is the monetary value associated with the transaction that involves Tom? Have you conducted business with Tom in the past? If so, what were the results? None of these attributes directly measures trust, but they can provide objective values that can be used to calculate a level of trust. It is ultimately up to the decision maker to determine if the proxy attributes can adequately be used to measure trust. This means that the decision maker must be able to understand the implication and extent that the proxy attributes relate to the subjective notion of trust. This may be difficult to transfer from one person to the next. This is not to say that a set of relationships cannot be made for the proxy attributes and the original attribute, but in complex problems this can be difficult.

Proxy attributes can be used to calculate a level of trust, but the final decision on trust is still a subjective one. However, this does not differ from making decisions based on objective attributes. The scalar values assigned to objective attributes are computed subjectively, although historical examples can provide highly accurate measurements. For example, a salary of $60,000 may statistically be the level at which loan defaults drop

to an acceptable level for a bank to automatically grant a loan. Additionally, given the same objective attributes, people will arrive at different conclusions, because decision making is a personal subjective act that involves experience, knowledge, attitudes, political consideration, and degrees of risk aversion. The goal is to provide as much objective information as possible so that a decision can be justified.

F. ASSESSMENT

It is the author's opinion that the trust models presented do meet the criteria for formal analysis, but they will be difficult to incorporate into a practical application. All of the models define the trust problem, assign values, choose alternatives, evaluate the alternatives by measuring the attributes, and then selecting the best alternative. They also all use commonly acceptable methods of measuring subjective attributes. However, each of the models evaluated contain flaws that complicate their being used effectively in conjunction with automated security protocols.

The models we evaluated follow formal analysis procedures, but they are difficult to apply to practical security protocols, such as X.509 and SPKI. The Reiter and Stubblebine model requires a paradigm shift in how certificates are used. PGP and the models proposed by Abdul-Rahman and Hailes, and Essin are not comprehensive enough to provide reliable measures of trust. In contrast, the Josang model is very comprehensive, but implementing it within the DoD PKI system will require a great deal of effort in order to agree on the PKI requirements, gain cooperation among the various participants, and standardize the system.

Another challenge these models pose is that they require a considerable amount of user input. The user must determine the scope of the trust decision, decompose the problem, assign values to trust attributes, and calculate costs and trust paths. The security protocols will need to apply the model to every transaction. The need for user input would significantly slow the processing of these protocols, and would require a prohibitive amount of time on the part of the user. None of the models evaluated can be fully automated in their current form.

THIS PAGE INTENTIONALLY LEFT BLANK

VIII. CONCLUSION

A. REQUIREMENTS TO IMPLEMENT JOSANG'S MODEL

To implement the Josang model with the DoD PKI system, a working group representing all of the agencies concerned, should be formed to determine methods of operation, standardization, training requirements, and interoperability needs. This group will also have to consider the needs and requirements of other governmental agencies, our Allies, and commercial partners because at some point, the PKI system will have to expand outside of the DoD domain.

1. Standardization

As discussed in chapter 4, one of the problems with the Josang model is that the subjective index is ambiguous. Josang does not provide confidence parameters to assist the user in assigning values. If the DoD is going to utilize Josang's trust model with its PKI system, a subjective index with discrete values in conjunction with his model is necessary to resolve some of the ambiguity in assigning values. For example, strong belief may have a value between .95 and .80. Without a measurement or criteria to use in the assignment of values, the ambiguities introduced by each person's subjective interpretations of the metric will affect the input values and how the output values are utilized.

To implement the Josang model, the DoD must develop a standardized subjective index with discrete values. Everyone utilizing the PKI system must understand and be able to apply the subjective index when assigning their trust values. This index must be applied to all of the trust values that a user assigns. The indexes can be the same, or they can differ depending upon the opinion that the user is forming. Indexes will have to be developed to assist in assigning values when evaluating a CA's public key, an entity's recommendation trustworthiness (colleague, third party, or inspection team), or when evaluating a CA's recommendation trustworthiness. A subjective index must also be used by the CA to measure its belief in the identity of a person. Finally, the resulting

output of the trust model must be measured against a subjective index to assist the user in making a trust decision.

In standardizing the subjective index, the DoD must decide on the level of granularity that is required (e.g., the index could be from 1-10, 1-100, or 1-1000) in the index. Should the scale be grouped into categories of good, marginal, and bad, or does the system require finer granularity? This may depend on the applications that are being used. If the applications involve high levels of security or their use is risky in nature, then the user may need very fine granularity. The DoD working group must decide on an index that will satisfy the majority of applications. The index will probably need a fine granularity, so it can apply to both high and low security concerns, but the group will have to decide that level of granularity.

The working group must also decide on how they want to implement the model. Chapter 6 offered a number of possible implementation techniques, but the working group must decide on an implementation approach. The method of implementing the model will drive the software requirements and the training that is required. Does the group want to use public key rings to store opinions on the recommendation trustworthiness of third parties, or do they want the user to manually enter that value for every decision? How does one person pass a recommendation on a certificate to another person? The group must decide whether they want to use extension fields in the certificate, or whether they want to use hash values appended to the certificate. Should the DoD convince browser manufacturers to include the DoD public key in their browser like other commercial CAs?

The working group must also decide on whether it wants to use a third party to inspect the CA. If it wants the CA inspected, it must decide how it wants the inspector's opinion on the trustworthiness of the CA published (e.g., incorporated in the certificate, published on a bulletin board). They must also agree on the criteria that will be used to evaluate the CA. How will civilian CA's be inspected? Will they submit to inspection? What agency will conduct the inspection, and are they qualified? Will our Allies allow a DoD inspection team to evaluate their CA? Will we allow our Allies to inspect our

system? Will we allow civilian agencies to inspect our CA? Can a cross certification of inspection agencies work?

2. Software Requirements

Although the software requirements will depend on the final system configuration and implementation plans developed by the DoD working group, some requirements are already known. The CA must be able to write an opinion into the certificate that it develops. The CAs already have the necessary software, but the use of the extension fields should be coordinated with the PKIX working group, which is responsible for standardizing the X.509 protocol.

If the DoD working group wants recommenders to append their opinions to certificates, or write and hash a recommendation within a certificate's extension field, then the DoD will have to develop and disseminate that software to all users. This includes colleagues of the user as well as the formal organizations that may be involved in inspecting CAs.

If the working group mandates the use of key rings, the key rings should hold private and public keys. Additionally, the public key ring should have a key recommendation field that contains the user's recommendation trustworthiness opinion on the public key holder. The key ring should also contain a key trust field to represent the user's trust in an entity's public key. A field is also needed to hold the recommendation opinions that recommenders give the user. If multiple recommendations are made, these values will have to be considered when calculating the final trust in the certificate. The final field should contain the user's opinion on the key-to-name binding of the certificate in question and certificates held in the key ring. The user should be able to modify any of the values at any time.

If key rings are not used, then the values necessary to compute the final trust value will have to be entered each time a transaction with a new certificate occurs. The program can extract the CA's recommendation from the certificate, but depending upon the methods used, the user must enter trust values for recommendation trust and trust in an entity's public key.

A GUI interface will have to be developed to allow the user to interact with the program. In addition to collecting the user's trust opinions, the GUI will also have to identify the type of transaction that is needed to determine a user's overall trust opinion on a certificate. If the user relies on multiple people to recommend a CA's recommendation trustworthiness, the GUI must be capable of differentiating that transaction from a transaction involving a recommendation from a third party. Temporary files may be needed to collect recommendations from multiple sources for later inclusion into the model. The user must also describe the actors and paths involved in the trust decision, so the GUI can prompt the user for the proper information.

The software to run Josang's trust model needs to be developed. All of the subjective logic formulas must be incorporated into software that interfaces with the X.509 certificate extension fields, a GUI interface, and possibly a key ring. Given the trust paths, actors and input from the user, the software should be able to compute a user's trust opinion. If information is needed, or inadequate, the user should be prompted with an error message.

3. Training

When the standardization issues and software requirements are completed, the users of the system must be trained. The DoD employees and anyone outside of the DoD domain that will also be using the DoD trust model PKI system must be trained on the operation of the system. They must understand the security risks associated with the Internet and the rationale behind trust models.

Training is needed on how to use the system, public key cryptography, PKI, trust chains, recommendation trust, subjective logic operators, and information needed to determine a trust opinion. Training will also be needed on how to interpret and use the system's output value to form a final trust decision.

The end users may not use the system properly unless they can see the benefits of using the system and the consequences of not doing so. Training should be geared to show how end users can benefit from using the system.

The CAs will also have to be trained on the use of the system. If they are cross certifying another CA, then they must be able to apply the evaluation techniques and know how to use the checklists in order to determine another CA's recommendation ability. They must also understand the standards that they will be held accountable for. If an outside agency is inspecting a CA, then they must both understand the evaluation criteria. Additionally the CA's must understand the subjective index used to measure the trust in a person's identity.

There will need to be a centralized agency responsible for administering the system (e.g., NSA, or a consortium) that will be able to act as an intermediary if an inspection team and a CA disagree on the interpretations of various evaluation criteria. This necessitates the requirement for a knowledgeable staff at NSA that will have the ability to make policy judgments.

Training materials, staff, and facilities will have to be identified, budgeted and developed before successful implementation of the PKI system. Additionally, issues such as who will fund and train our Allies and business partners will have to be resolved. If we are forcing a contractor to use the DoD PKI system, is the DoD responsible for training costs, or is the contractor? Who will fund a third party inspection organization?

4. Cooperation

Any organization operating outside of the DoD domain that wishes to interface with the DoD PKI system must abide by the standards implemented by the working group. However, the working group must attempt to incorporate the needs and desires of the DoD's business partners, Allies, and fellow government agencies in their decisions.

In some cases, to satisfy the requirements set forth by the working group, CAs will have to adopt new business practices, adopt new software, and open their operation to an outside inspection team. CAs will also have to be flexible enough to adapt to changing requirements established by the DoD. As PKI technology continues to evolve, requirements will evolve as well.

The DoD must realize the efforts that CA's must make to become interoperable with the DoD PKI system. If the requirements are too stringent, are difficult to

implement, or are too costly, then the DoD will have a system that nobody outside of the domain will use.

5. Automation

The goal of most security products is to appear invisible to the user, but still perform the functions that are required. It is possible to fully automate Josang's model within the DoD PKI system; however, the user will have to assign the CA full trust. When the user installs the PKI program on his or her computer and loads the CA's public key, the user will have to have complete trust in the validity of the CA's public key and the CA's recommendation trustworthiness. When a certificate is received, the system will validate the certificate using the CA's public key and will read the CA's trust opinion on the identity of the certificate's owner. Unfortunately, a fully automated system that requires complete trust in the CA violates the rationale for using Josang's trust model.

If a key ring concept is used, it is possible to partially automate Josang's trust model within the DoD PKI system. A user can enter opinions on the CA's public key and recommendation trustworthiness in the appropriate fields in the key ring. When a user receives a certificate, these values could then be used with the information in the extension fields to generate a trust value output. This value could be measured against parameters defined by the user to either accept the certificate, or display a warning screen. This system would also require the user to enter opinions regarding the validity of friends' and organizations' public keys and their recommendation trustworthiness. These values could then be used by the user to generate an opinion on the CA's recommendation trustworthiness. This system would have to be flexible enough to allow the user to modify opinion values in any of the fields.

B. FUTURE WORK

The Internet has created an opportunity for organizations to gather more quantitative and qualitative information for decision makers. The ability to analyze information faster and more efficiently than the competition permits organizations to better position themselves in the marketplace so as to react quickly to changes in the

business environment. As organizations become more reliant upon the Internet to exchange information, the need for trust models and trust management systems will increase. Organizations will need the greater security and better authentication techniques the trust systems offer. Possible topics for further research include the following:

1. Distributed Databases

On-Line Analytical Processing (OLAP) tools utilize general purpose Web browsers to facilitate the analysis of large amounts of data from distributed databases. The information utilized by the OLAP technologies can be incorporated from a number of sites in different formats. Military applications incorporate OLAP technology to combine sensor information into databases used for command and control. As technology continues to advance, it will not be long before OLAP programs will create intelligent agents that can search the web for information sources. However, before these agents are implemented, issues such as interconnectivity among the agents and information sources, data overflow, data validity, and security must be addressed. Trust management techniques can be used to address the security concerns involved with collecting information over an inherently untrustworthy medium.

2. Developing Subjective Indexes

Currently there is no subjective index to use with the Josang trust model. More research needs to be conducted concerning the use of indexes with trust models. What are the various types of indexes that can be used? What level of granularity is necessary? Are the discrete scales used in PGP better than assigning numerical values? Should the index be modeled visually like Josang's opinion triangle, where the values of belief, disbelief, and uncertainty correspond to the sides of a triangle? (Josang, A., 1997) Can proxy attributes be used to measure trust? If multiple indexes are needed to evaluate the various attributes of trust, how can the indexes be combined to form one opinion, without giving too much weight to any single attribute? Can an output formula compensate for multiple indexes? Research can also be done on trying to apply multi-attribute utility theories and software to trust models.

3. Inspecting Certificate Authorities

If third-party organizations are going to evaluate CAs, then an inspection criterion must be developed. This may consist of a check-off list that assigns point values to various tasks that the CA performs (e.g., certificate revocation, key security, or identity authentication). The evaluation should cover all of the tasks that an ideal CA should perform. Each task should have an index assigned that will allow the investigators to determine a final trust opinion. Research in this area can also focus on determining the information an inspector must have to properly evaluate the CAs. This research can be extended to cover the evaluation criterion that a CA must use to determine if another CA is trustworthy enough to perform cross certification.

4. REFEREE Trust Management System

In order for intelligent agents to interact with the Internet, in a secure manner, a methodology must be developed for identifying and authenticating web sites and their information. One of the methods to accomplish this is to add labels to the web sites that contain their certificates and outline the information contained in each site.

Utilizing the aforementioned labels, organizations might be better able to evaluate the information they are receiving from the Internet. However, a trust management system would probably still need to be implemented to ensure that the information gathered from the Internet met with certain user-defined trust criteria specific to that of a user or organization.

In order to implement a trust management system, all of the principals (entities involved) should be identified. The REFEREE trust management system utilizes digital certificates to identify the principals, but they assume that certificates assure authentication. Utilizing trust models with the PKI system could make the REFEREE system more secure.

5. Applying Trust Models to Downloaded Code

When downloading information from a Web site, how does the agent know whether the information contains malicious code? Additionally, if a client downloaded

Java applets or Active X, how does the client know whether the mobile code is malicious until it is too late to prevent the malicious code from executing?

The user must form an opinion concerning the extent to which he or she trusts the developers of the downloadable program and the web site that is distributing the program. However, to form a trust opinion, a user must be able to analyze the risks, and obtain enough information to make an educated decision. The user must also have enough practical knowledge (e.g., programming knowledge and security knowledge) to be able to make an informed decision. This can be difficult when dealing with complex technical issues. Trust models can be applied to give the user the necessary information to make a trust decision.

6. New Trust Model

None of the models discussed in this thesis are flawless. One thread of research could center on the variables and metrics used within the models. PGP and Josang's trust models rely on the user to determine the trust values he enters into the model (e.g., the user assigns a recommender trust value of (.9, .05, .05) to Bob). The trust decision is made outside of the model, so it is difficult for an outsider to audit a decision. If the DoD adopts Josang's model, and third party inspection teams are used, the opinions formed by an inspection team must be justified. In order to defend their opinion, the team must collect and evaluate information based on a standardized inspection criterion. Can a model be developed that would incorporate the inspection criterion into the model itself? The model should be able to question the user concerning trust attributes that could be used to form a trust opinion. All of the models evaluated contained different trust variables. Can a model be developed that incorporates the best variables from the models, while eliminating the individual model's weaknesses? Additionally, what is the minimum set of variables needed to support a particular level of trust functionality?

7. Trust Model Applications

Trust models can be utilized in numerous applications. Currently, PGP contains the only formal trust model that is currently in use. Additional research can explore

applications that could benefit from utilizing trust models. One area that can benefit from trust models is in the field of knowledge management.

Subjective logic is capable of combining the opinions of numerous people together to form one opinion. Expert systems attempt to capture the knowledge of the most experienced workers and automate the decision methodologies that they utilize. This is most often in the form of heuristics, or rule based decision methods. However, these systems have difficulty collecting information from numerous experts, because they do not always agree on a problem solving methodology. Subjective logic can take the experts' opinions on a problem and combine them into a single opinion that could then be incorporated into an expert system.

Trust models can also be used in intrusion detection systems. One intrusion detection technique is to use profiling. Information is collected on a user to develop a profile of his attributes, work habits, and information that is commonly needed to perform his job. Some of the information includes typing speed, the time he arrives at the office, and files accessed. The system uses the profiles to search for deviations from the profiles that might indicate that an intrusion has taken place. If a user never works on the weekend, then the intrusion detection system will generate an alert if his computer is accessed over the weekend. The use of a trust model allows the program to combine the values of various profile attributes to determine if an intrusion is occurring. Trust models can also take sensor information on the same abnormality and combine the values to produce a more accurate measurement.

8. Implementation Issues

The weaknesses in current methods of authentication will necessitate some form of trust model, or trust management system. Implementing current trust management systems will be difficult, but as the PKI systems continue to evolve, some of the difficult interoperability and management problems will have to be addressed. Follow-on research can focus on the management and coordination issues necessary to implement a trust model in the DoD PKI system.

9. Red Teams

In order to encourage confidence in cryptography systems, red teams are developed internally and sometimes externally to exercise the algorithm to find weaknesses. A red team can also be developed to test the trust model and the DoD's PKI system. Follow-on research can focus on developing tests that will evaluate the proficiency of the CAs, determine the strength of the subjective logic algorithm, determine the best method of distributing the CA's public key, evaluate the system's resistance to outside attack, and evaluate the best architecture in which to implement the trust model.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX. GLOSSARY

Address Spoofing: Altering the TCP/IP packets to make it appear that the message came from a source other than the originator.

Authentication: The process used to ascertain the identity of a subject.

Availability: Ensures that computer assets are fully operational when needed.

Back Door: An undocumented access code or procedure for accessing information.

Bridge CA: Instead of the CAs cross certifying with each other, they cross certify with a third party "bridge CA" that acts as an intermediary between CAs.

Certificate: A data structure that securely links an entity with its corresponding public key.

Certification Authority (CA): The component of the public key infrastructure that is responsible for issuing, revoking and certifying public keys.

Certificate Revocation List (CRL): A list of certificates that have been cancelled before their expiration date.

Ciphertext: The output of an encryption algorithm, or the encrypted form of a message.

Confidentiality: Ensures that information within a computer or transmitted can only be read by authorized personnel.

Cryptography: The branch of cryptology that deals with the algorithms that encrypt and decrypt messages or files to provide security and/or authenticity. (Stallings, W., 1999)

Digital Signature: An authentication mechanism that utilizes public key cryptography to guarantee the source and integrity of a message.

Domain: The logical realm over which a CA determines policy.

FORTEZZA: "FORTEZZA®" is a registered trademark held by the National Security Agency. It describes a family of security products. The FORTEZZA crypto card started as a low cost security device for the Defense Message System. However, the card was designed to be a general purpose cryptographic "co-processor" that can be used in numerous applications. The DoD class 4 PKI system uses FORTEZZA standards.

Hackers: People who abuse information systems or use them to commit criminal acts.

Hash Function: A function that combines a bit string with a secret key to generate a fingerprint of the message. The recipient of the message uses the same key to generate a hash value of the message and compares the two hash values. If they are the same, the message's integrity is valid.

Integrity: Only authorized personnel can modify computer assets or transmissions.

Key: A string of bits used in encryption algorithms to encrypt plaintext and decrypt ciphertext. The string's length depends upon the type of algorithm used.

Local Registration Authority (LRA): The person or organization that is responsible to be CA for properly identifying an entity seeking a certificate.

Lightweight Directory Access Protocol (LDAP): The defacto standard for accessing directory systems.

Nonce: An identifier or number that is used with authentication techniques to combat the man-in-the-middle attack.

Non-Repudiation: A message is sent such that the identity of the sender and the integrity of the message are strong enough to prevent that party from later denying that the transaction ever occurred.

Plaintext: The message that is to be encrypted, or the message that is recovered from decryption.

Pretty Good Privacy (PGP): A public-key cryptography program that was developed primarily by Phil Zimmerman in 1991.

Private Key: One of two keys used in public key cryptography. The private key is known only to the user and should be kept secret. Only the user should have the private key. The private key decrypts the corresponding public key.

Public Key: One of two keys used in public key cryptography. The public key is made available to everyone. The public key can decrypt its corresponding private key to verify authenticity (digital signature).

Public Key Cryptography: Cryptography that uses a pair of related keys to perform cryptography. When the keys are generated, one is designated the "private key", which is kept secret and the other key is the "public key", which is available to everyone. Public key cryptography is also called asymmetric cryptography.

Public Key Infrastructure (PKI): The key management system that ensures public keys are safely, efficiently, and conveniently delivered to the system that needs them.

Registration Authority (RA): In many cases the actual identity verification is delegated from the CA to another organization called registration authority (RA).

Root Certificate Authority: The most trusted entity in a hierarchical PKI domain. It is responsible for establishing and maintaining the PKI domain. It establishes the policy, issues the certificates and delegates responsibilities to lower level CAs or LRAs. It is the trust anchor.

Subjective: The evaluation of an object or occurrence is unique to each person.

Subjective Logic: It consists of a set of algebraic operators. It can be called a calculus for uncertain probabilities.

Symmetric Cryptography: The same key that is used to encrypt the message is used to decrypt the message.

Transitivity: In the context of trust, in order for trust to be transitive in a trust path, trust must be valid for each member in the path. For example, Bob trusts Sue, and Sue trusts Tom, transitivity assumes that Bob trust Tom.

Trojan Horse: An innocent looking program that has additional malicious functions.

Trust Anchor: The CA that is fully trusted by a user. This means that the user has complete trust in the CA's public key.

Trust Models: They attempt to automate the logic, variables, and thought processes that a human performs when making a trust decision.

Trusted Path: The verification path that a user must take to verify a certificate with a trusted CA.

Virus: A self- replicating computer program. A virus is often malicious code embedded in an executable program.

Worm: A self-replicating program, but unlike a virus it does not need a host to propagate, it is designed to spread on its own. It is malicious in that it performs a denial of service attack.

X.509 Standard: The standard that defines the structure and functionality for certificates and CRLs.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Abdul-Rahman, A., and Hailes, S., "A Distributed Trust Model," NSPW '97. Proceedings of the Workshop on New Security Paradigms Workshop, pp. 48-60, 1997.

Barton, D., "Design Issues in a Public Key Infrastructure (PKI)," [http://www.csu.edu.au /special/auugwww96/proceedings/barmoroco/barmoroco.html], 1996.

Booker, R., "Practical PKI," *Messaging Magazine*, September/October, 1999.

Cabletron Systems, "Public Key Infrastructure (PKI)," [http://www.Cabletron.com/vpn/ VPNpki.htm], 10 June 1999.

Cheswick, W. and Bellovin, S., *Firewalls and Internet Security*, Addison-Wesley Publishing Company, 1994.

Chu, Y., "Trust Management for the World Wide Web," Master's Thesis, Massachusetts Institute of Technology, Boston, Massachusetts, 13 June, 1997.

Chu, Y., Feigenbaum, J., LaMacchia, B., Resnick, P., and Strauss, M., "REFEREE: Trust Management for Web Applications," [http://www.research.att.com/~bal/papers/www6-referee/www6-referee.html], 1997.

Fearnley-Sander, D., "Hermann Grassmann and the Prehistory of Universal Algebra," *American Mathematical Monthly*, v.89, pp.161-166, 1982.

Ford, W. and Baum, M., *Secure Electronic Commerce, Building the Infrastructure for Digital Signatures and Encryption*, Prentice Hall PTR, 1997.

Essin, D., "Patterns of Trust and Policy," Proceedings of the Workshop on New Security Paradigms Workshop, NSPW '97, pp. 38-47, 1997.

Ford, W., "Public-Key Infrastructure Interoperations: Some Pragmatics," *Messaging Magazine*, September/October, 1997.

Gerck, E., "Towards Real-World Models of Trust: Reliance on Received Information," [http://www.mcg.org.br/trustdef.htm], 1998.

Hansen, A., *Public Key Infrastructure (PKI) Interoperability: A Security Services Approach to Support Transfer of Trust*, Master's Thesis, Naval Postgraduate School, Monterey California, September, 1999.

Hombeck, R., "The Troubling Truth About Trust on the Internet," *EDI Dorum, The Journal of Electronic Commerce*, v. 10, no. 4, pp. 59-70, November 1998.

Josang, A. "A Logic for Uncertain Probabilities," unpublished, September 1999.

Josang, A., "A Metric for Trusted Systems," *Proceedings of the 21st National Security Conference*, NSA, 1998.

Josang, A., "A Subjective Metric of Authentication," *Proceedings of the 5th European Symposium on Research in Computer Security (ESORICS'98)*, Springer-Verlag, 1998.

Josang, A., "An Algebra for Assessing Trust in Certification Chains," *Proceedings of the Network and Distributed Systems Security (NDSS'99) Symposium, The Internet Society*, 1999.

Josang, A., "Artificial Reasoning with Subjective Logic," *Proceedings of the Second Australian Workshop on Commonsense Reasoning*, 1997.

Josang, A., "Trust-Based Decision Making for Electronic Transactions," *Proceedings of the Fourth Nordic Workshop on Secure Computer Systems (NORDSEC'99)*, Stockholm, 1999.

Josang, A., "Prospectives for Modeling Trust in Information Security," *Proceedings of the 1997 Australasian Conference on Information Security and Privacy*, Springer, 1997.

Kabay, M., *The NCSA Guide to Enterprise Security*, McGraw-Hill, 1996.

Keeney, R., *Value-Focused Thinking a Path to Creative Decision Making*, Harvard University Press, 1992.

Keeney, R., and Raiffa, H., *Decisions with Multiple Objectives*, Cambridge University Press, 1993.

Khare, R., and Rifkin, A., "Trust Management on the World Wide Web," [http://www.firstmonday.dk/issue3_6/khare/], June 1998.

Khare, R., and Rifkin, A., "Weaving a Web of Trust," [http://www.cs.caltech.edu/~adam/local/trust.html], 30 November 1997.

LaMacchia, B., "The Digital Signature Trust Management Architecture," [http://www.research.att.com/~bal/dsig/tma-design.htm], 10 January 1997.

McCullagh, A., "The Establishment of 'Trust' in the Electronic Commerce Environment," [http://www.acs.org.au/president/1998/past/io98/etrust.htm], 7 November 1998.

Myers, A., and Liskov, B., "A Decentralized Model for Information Flow Control," *ACM SIGOPS Operating Systems Review*, v. 31, no. 5, pp. 129-142, December 1997.

Perlman, R., "An Overview of PKI Trust Models," *IEEE Network*, pp. 38-43, November/December 1999.

"Public Key Infrastructure Roadmap for the Department of Defense," Version 2.0, Revision C, Department of Defense, May 6, 1999.

Reiter, M., and Stubblebine, S., "Authentication Metric Analysis and Design," *ACM Transactions on Information and System Security*, v. 2, no. 2, pp. 138-158, May 1999.

Render, B., and Stair, R., *Quantitative Analysis for Management*, 6th ed., Prentice Hall, 1997.

Stallings, W., *Cryptography and Network Security Principles and Practice*, 2d ed., Prentice Hall, 1999.

Stokey, E., and Zeckhauser, R., *A Primer for Policy Analysis*, W. W. Norton and Company Inc., 1978.

"United States Department of Defense X.509 Certificate Policy," Version 2, Department of Defense, [http://csrc.nist.gov/pki/twg/dod.htm], March 1999.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
   8725 John J. Kingman Road, Ste 0944
   Fort Belvoir, VA 22060-6218

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Road
   Monterey, CA 93943-5101

3. Dean Dan Boger 1
   Code IW
   Naval Postgraduate School
   Monterey, CA 93943-5118

4. Professor James Bret Michael, Code CS/Mj 1
   Naval Postgraduate School
   Monterey, CA 93943-5118

5. Professor Rex Buddenberg, Code SM/Br 1
   Naval Postgraduate School
   Monterey, CA 93943-5118

6. LCDR Leonard T. Gaines 2
   945 Fletcher LN Apt #315
   Hayward, CA 94544

7. CAPT Rhea 1
   Naval Supply Systems Command
   5450 Carlisle Pike
   Building 309, Room 305
   Mechanicsburg, PA 17055-0791

8. CDR Ian Anderson 1
   1412 Essex Cir
   Ridgecrest, CA 93555

9. LCDR Anthony Hansen 1
   Naval Research Laboratory
   Bldg. 259/Code 9110
   4555 Overlook Ave.
   Washington, D.C. 20375

10. Mr. Terry Mayfield 1
Computer and Software Engineering Division
Institute for Defense Analysis
1801 North Beauregard Street
Alexandria, VA 22311-1772

11. Professor John McHugh 1
SEI/CERT
4500 5th Avenue
Room 4420
Pittsburgh, PA 15213-3890

12. Professor Audun Josang 1
DSTC Pty Ltd
Level 7, GP South (Bldg 78)
The University of Queensland
Brisbane, QLD 4072
Australia